

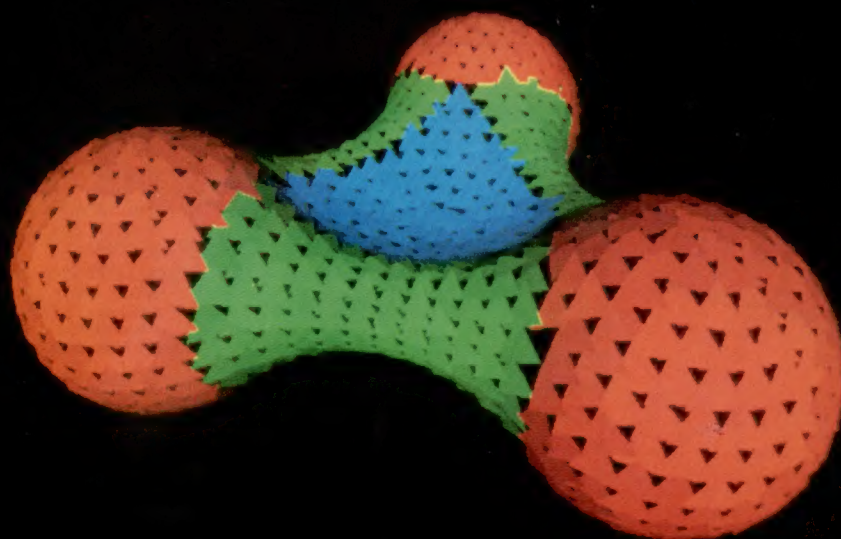
abc

N° 41

COURS
D'INFORMATIQUE
PRATIQUE
ET FAMILIALE

INFORMATIQUE

VECTRIX CORPORATION



THREE ATOMS Courtesy of GREG ABRAM, University of North Carolina at Chapel Hill

Le gang des réseaux

D'une pierre deux coups

Dans la vallée des Trolls

Une nouvelle ÈRE

EDITIONS
ATLAS

M6062-41-12F

85FB-3,80FS-\$1.95



Gang des réseaux

Les réseaux informatiques peuvent être de dimension nationale, ou d'une échelle plus réduite. Voyons comment il est possible de constituer un réseau à partir d'ordinateurs domestiques.

Un réseau est un ensemble d'ordinateurs reliés les uns aux autres pour partager des données et de l'équipement. Cependant, chaque ordinateur a son propre système d'exploitation et ses propres « protocoles » (procédures, règles de formatage, etc.) pour communiquer avec l'extérieur. Pour des raisons de compatibilité, les « stations » individuelles d'un réseau doivent être des ordinateurs semblables, de la même marque. Pour illustrer notre propos, supposons qu'un groupe de personnes possède cinq ordinateurs qu'ils désirent relier à une seule imprimante. Notre groupe doit être en mesure d'envoyer de l'information à partir de l'un ou de l'autre des terminaux. Que se passe-t-il si deux ou trois ordinateurs essaient d'imprimer du texte en même temps? Mieux encore, qu'arrive-t-il si l'ordinateur 3 a du texte à imprimer mais doit continuer à travailler pendant l'impression? Pour résoudre ces problèmes, nous devons installer un sixième ordinateur, nommé « serveur » d'impression. Cette machine est réservée au contrôle du débit des données vers l'imprimante, et ne peut donc servir à rien d'autre. Ce serveur stocke les documents par ordre de priorité. Dès qu'une portion

de texte est envoyée d'un ordinateur au serveur, l'ordinateur peut se livrer à d'autres tâches.

L'utilisation d'une telle machine est essentielle dans un réseau, puisque c'est par l'intermédiaire du serveur que l'information peut être partagée. Par ailleurs, certaines applications nécessitent la présence d'un serveur de fichiers pour gérer le partage des lecteurs de disquettes et pour contrôler le débit de données d'un ordinateur à l'autre.

Il faut ensuite créer une liaison entre les ordinateurs. Pour cela, on installe un câble — soit une paire torsadée ou un câble coaxial — d'une machine à l'autre. Bien qu'il existe plusieurs dispositions de réseaux, le principe demeure le même. Pour établir la connexion, il est généralement nécessaire de munir chaque ordinateur d'une interface de réseau. Il peut s'agir d'une simple connexion RS232 ou d'une carte de circuit imprimé enfichable. De plus, le poste serveur a besoin d'une unité de stockage d'une capacité suffisante pour gérer tout le débit de données. Le terminal serveur a aussi besoin d'une RAM suffisante pour gérer le réseau.

C'est le logiciel utilisé par un ordinateur qui détermine son efficacité, et cela est particulière-



Interface 1

Les réseaux ne nécessitent pas toujours des kilomètres de câbles et des équipements très coûteux. Les Spectrum, équipés de l'Interface 1, peuvent communiquer entre eux et partager des Microdrive et une imprimante, ce qui constitue un système très économique.

(Cl. Paul Chave.)



ment vrai dans le cas des réseaux. D'abord et avant tout, un logiciel doit couvrir le système d'exploitation et établir une connexion entre chaque ordinateur et le réseau. Connus sous le nom de « logiciel de gestion de réseau », il confie au serveur les opérations spécialisées de gestion de fichiers et de gestion de l'imprimante, ainsi que le contrôle du débit des données. En plus, le logiciel de gestion de réseau informe chaque ordinateur qu'il fait bien partie du réseau, qu'un serveur est connecté et lui indique le nombre d'ordinateurs faisant partie du système. Finalement, ce logiciel essentiel donne aux ordinateurs du réseau un protocole pour leur permettre de communiquer avec les autres.

Dès que le réseau est constitué, les ordinateurs individuels doivent posséder un programme ou un ensemble de programmes, pour leurs propres applications, qui sachent reconnaître le réseau et communiquer avec lui. Ce logiciel, écrit spécialement, peut être exécuté à partir d'une cassette, d'une unité de disquette ou par l'intermédiaire du serveur de fichiers. La complexité du logiciel est fonction de la tâche. Si l'ordinateur 1 de notre réseau exécute un pro-



Expérience partagée

Cette école a acquis 16 micros munis de moniteurs couleur, d'une imprimante, d'un lecteur de disquettes double et d'un logiciel de communication pour la somme de 180 000 F, ce qui est moins coûteux que d'acheter une imprimante et un lecteur de disquettes pour chaque ordinateur. La vitesse du réseau est telle que chaque terminal a l'impression d'être le seul à utiliser le lecteur de disquettes, même si 30 élèves travaillent simultanément; les terminaux peuvent avoir à se mettre en file d'attente pour utiliser l'imprimante. (Cl. Tony Sleep.)

gramme de traitement de texte et envoie les résultats à l'imprimante, indépendamment des autres ordinateurs, la seule modification à apporter à un programme standard de traitement de texte consistera à inclure des protocoles de réseau. Par contre, si les ordinateurs 2 et 4 désirent utiliser les mêmes données et si chacun veut consulter les résultats de l'autre, les choses deviennent plus compliquées. Ici, le logiciel d'applications (que ce soit un programme de traitement de texte, un tableur, une base de données, ou même un jeu) et le matériel doivent être en mesure de gérer des tâches multiples. En d'autres termes, l'UC doit pouvoir gérer plus d'une tâche à la fois et, simultanément, les communications provenant d'au moins deux autres UC.

Sinclair Research, par exemple, a travaillé sur ce problème. Une interface de réseau est intégrée dans l'unité complémentaire Interface 1 destinée au Spectrum. Cette unité se vend très bien, puisqu'elle est requise pour relier le Spectrum aux Microdrive. Dès qu'un nombre suffisant d'Interface 1 aura été vendu, il est probable que de nombreux programmes utilisant l'interface de réseau feront leur apparition.

Le dernier micro de Sinclair, le QL, a une interface similaire intégrée dans la version de base (le TO 70 de Thomson et l'Exelvision également), et cette interface devrait être compatible avec la version du Spectrum. Bien que cette interface soit plutôt sommaire, la popularité de ces machines devrait rentabiliser la production de programmes lui étant destinés. Les programmes de jeux sont évidemment les premiers candidats.

Plusieurs éléments sont nécessaires pour constituer un réseau vraiment fonctionnel. Nous devons d'abord avoir deux ordinateurs à relier. Deuxièmement, les ordinateurs ne doivent pas être trop éloignés, afin de pouvoir être reliés au moyen d'un câble. Cela signifie qu'ils doivent être dans le même immeuble. Finalement, il doit y avoir assez de « circulation » pour justifier l'établissement du réseau. Autrement dit, il doit y avoir un nombre suffisant d'utilisateurs qui désirent échanger des données plusieurs fois par jour ou partager un périphérique coûteux (comme une imprimante ou un lecteur de disquettes) pour amortir le coût du réseau. Il est encore possible, moyennant finance, de s'intégrer aux réseaux spécialisés, comme Transpac en France.

Si une petite quantité de données seulement doit être véhiculée au sein du réseau, il serait plus simple de le faire au moyen de disquettes ou de cassettes. De même, si le réseau n'est formé que de deux ou trois ordinateurs, il serait plus économique d'équiper chaque ordinateur de sa propre imprimante et de son propre lecteur de disquettes plutôt que d'investir dans l'établissement d'un réseau.

Par conséquent, à l'exception des jeux, les seules utilisations pratiques pour les réseaux d'ordinateurs domestiques ne s'adressent encore qu'aux petites entreprises et au milieu scolaire.



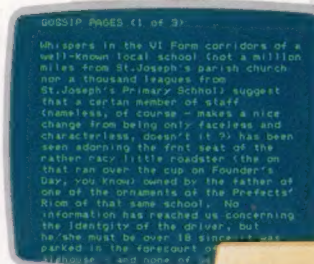
Boîtier d'horloge
Génère les signaux de synchronisation qui synchronisent toutes les communications du réseau.



RÉDACTEUR EN CHEF



Serveur de fichiers
Ce micro est dédié au pilotage du réseau. Les utilisateurs ont leurs propres mots de passe et leurs répertoires privés de disquettes. Il y a aussi un répertoire public de fichiers mis à la disposition de tous les utilisateurs.



RESPONSABLE DE LA PRODUCTION



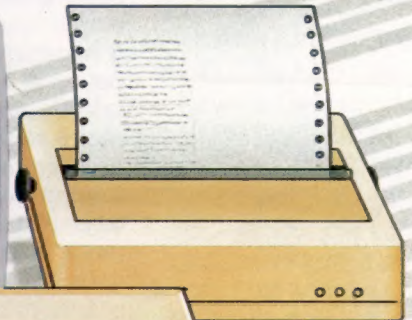
Coupleur acoustique
Transmet une information téléphonique au lecteur de disquettes simple par l'intermédiaire d'un terminal non connecté qui entre alors dans le réseau et transmet l'information.



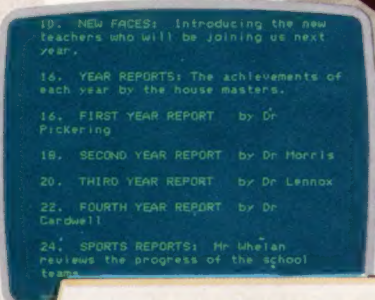
BOÎTE DE JONCTION



Contrôleur
Terminaison électrique des extrémités du bus de données pour empêcher toute détérioration du signal.



Imprimante
Sollicitée par tout terminal et contrôlée par une ROM dans l'un des terminaux, ce qui permet aux terminaux de continuer à travailler normalement.



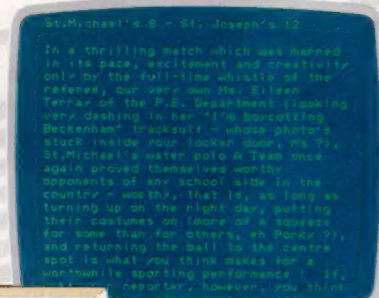
Identité du terminal
Chaque terminal a un numéro unique d'identification défini par des interrupteurs internes.



La presse informatisée

RÉDACTEUR
Les rédacteurs utilisent leurs terminaux comme systèmes de traitement de texte et sauvegardent leurs copies sur le lecteur de disquettes commun. L'équipe de production peut à tout moment visualiser les écrans des rédacteurs sur ses propres terminaux et éditer par la suite la copie finale à partir de fichiers sur disquettes. Le micro du rédacteur en chef pilote le réseau, et une copie finale est produite par l'imprimante. Des fichiers provenant de l'ordinateur d'une autre école sont transmis au micro responsable de la production. (Cl. Steve Cross.)

SECRÉTAIRE DE RÉDACTION



Accès aux terminaux
Tout terminal peut copier des messages venant de tout autre écran terminal ou envoyer des messages.



RÉDACTEUR



Tout est contrôlé

ÉLITE
 ABCDEFGHIJKLMNOP
 QRSTUVWXYZabcdef
 ghijklmnopqrstuv
 wxyz0123456789 !

"£\$%&'()*+,-./:;

<=>?@[\\]^_`{|}~

PICA

ABCDEFGHIJKLM
 NOPQRSTUVWXYZ
 abcdefghijklm
 nopqrstuvwxyz
 0123456789 !"£\$%&'()*+,-./:;

;<=>?@[\\]^_`{|}~

<=>?@[\\]^_`{|}~

<=>?@[\\]^_`{|}~

PICA GRAS EN ITALIQUE

ABCDEFGHIJKLM
 NOPQRSTUVWXYZ
 abcdefghijklm
 nopqrstuvwxyz
 0123456789 !"£\$%&'()*+,-./:;

;<=>?@[\\]^_`{|}~

;<=>?@[\\]^_`{|}~

;<=>?@[\\]^_`{|}~

ÉLITE AGRANDI

ABCDEFGHI
 IJKLMNOP
 QRSTUVWX
 YZabcdef
 ghijklmn
 opqrstuv
 wxyz0123
 456789 !

"£\$%&'()*+,-./:;

<=>?@[\\]^_`{|}~

<=>?@[\\]^_`{|}~

<=>?@[\\]^_`{|}~

<=>?@[\\]^_`{|}~

<=>?@[\\]^_`{|}~

<=>?@[\\]^_`{|}~

<=>?@[\\]^_`{|}~

PICA CONDENSÉ

EN DOUBLE FRAPPE

ABCDEFGHIJKLMN
 OPQRSTUVWXYZ
 abcdefghijklm
 nopqrstuvwxyz
 0123456789 !"£\$%&'()*+,-./:;

Caractères pointillés

Les imprimantes à matrice de points permettent des polices de caractères variées telles que Pica, Élite et Italique, et des frappes particulières telles que condensées, agrandies et « gras ». Tous les exemples ici proviennent d'une Epson FX-80.

Même les imprimantes à matrice de points les moins chères disposent d'une gamme d'« effets spéciaux ». Voyons comment ces effets sont obtenus et comment imprimante et ordinateur dialoguent.

Une imprimante à matrice de points peut produire autre chose que des listages. Un rapide coup d'œil au manuel utilisateur de l'imprimante vous donnera un aperçu des possibilités d'impression. Même l'imprimante à matrice de points la moins chère vous permettra de choisir la taille des caractères. Le texte est normalement tapé à raison de 80 caractères par ligne, mais ce nombre peut être accru en choisissant le mode « impression condensée » (qui utilise des caractères plus petits), ou diminué en choisissant le mode « impression agrandie ». L'espacement entre les lignes peut être également modifié.

L'imprimante que nous étudions en détail ici, l'Epson FX-80, est un bon exemple de machine dotée d'une vaste gamme de caractères d'impression. Le mode « gras » et le mode « italique » sont deux de ses caractéristiques. Mais le plus remarquable, peut-être, est sa capacité à modifier tout caractère présent dans la mémoire de l'imprimante, ce qui permet des caractères typographiques propres à certaines langues ou encore des symboles scientifiques. Avant de voir comment ces caractéristiques sont rendues possibles, voyons d'abord comment une imprimante reproduit un listage de programme.

La communication entre l'ordinateur et l'imprimante varie d'une machine à l'autre. Le Dragon par exemple utilise une simple variante d'une commande LIST, LLIST, pour demander à l'imprimante une copie d'un programme. D'autres machines supposent de passer par des « canaux » ou « séquences » pour accéder à l'imprimante. Comme la méthode varie toujours d'une machine à l'autre, le mieux est de consulter le manuel utilisateur de votre ordinateur.

Une fois la communication établie entre les deux machines, votre premier essai d'impression peut se révéler décevant : une seule ligne incompréhensible de caractères qui s'« écrasent » les uns les autres, par exemple, ou des lignes démesurément espacées. L'origine commune à ces deux dysfonctionnements vient de la différence entre un caractère « saut de ligne » et le « retour-chariot ». Lorsque votre ordinateur envoie à l'imprimante une ligne de texte, elle envoie à la suite un « retour-chariot » qui replace la tête de l'imprimante à la ligne et à la marge gauche de la page. Certains ordinateurs font ainsi avancer la page d'une ligne, alors que d'autres supposent que l'imprimante s'en charge. Pour compliquer encore les choses, la plupart des imprimantes disposent d'un com-

mutateur interne qui décide d'un saut de page ou de son absence.

Outre l'impression de listages de programmes, l'imprimante est aussi un périphérique pour « sortir » des données vers un support papier au lieu de l'écran. Ici encore, la méthode exacte varie selon l'ordinateur. La commande BASIC standard est LPRINT, elle est utilisée par Oric et Spectrum. Sur un Commodore 64, OPEN1,4 suivi de PRINT 1, «BONJOUR», imprimera le mot «BONJOUR». Avec un Dragon, la même tâche sera accomplie par PRINT -2, «BONJOUR». Le BBC utilisera VDU2 suivi de PRINT «BONJOUR» et de la commande VDU3. Les exemples de programmation que nous donnons ici utilisent LPRINT, aussi vous aurez peut-être à les adapter à votre machine.

Étiquettes d'adresses

```
10 LPRINT "M JEAN LAFLEUR"
20 LPRINT "7 GRANDE RUE"
30 LPRINT "MOULINS"
40 LPRINT "ABC 123"
50 FOR I=1 TO 7
60 LPRINT
70 NEXT I
100 GOTO 10
```

Ce listage est celui d'un programme simple destiné à produire des étiquettes d'adresses. Il existe du papier en continu sur lequel figurent des étiquettes autocollantes. Des perforations latérales à cette liasse permettent de disposer ce type de papier dans le chargeur de l'imprimante. Comme ce programme n'utilise pas de codes spéciaux de commande, le programme fonctionnera sur n'importe quelle imprimante. Tel qu'il se présente ici, le programme imprimera indéfiniment le même nom et la même adresse. Si vous voulez le modifier, vous devrez entrer des noms et des adresses différents, ou encore les faire lire depuis un fichier de données. La boucle FOR...NEXT entre les lignes 50 et 70 imprime sept lignes vierges et permet de positionner convenablement la tête de l'imprimante sur chaque étiquette. Le nombre exact de lignes blanches devra, éventuellement, être adapté à votre machine.

Notre programme convient tout à fait à l'impression d'étiquettes. Pour imprimer des textes plus complexes, nous allons devoir utiliser certains des effets spéciaux que nous avons mentionnés précédemment. Ils sont générés en envoyant des codes de commande appropriés à l'imprimante ainsi qu'aux caractères normaux du texte.



Outre qu'elle code chaque caractère du clavier, la table des caractères ASCII comporte un jeu de caractères « invisibles » qui n'impriment ou n'affichent rien. Ces codes servent notamment à activer les effets spéciaux de l'imprimante : pour la table ASCII standard, ce sont les quatre codes 17, 18, 19 et 20 réservés en tant que commandes de périphériques. Malheureusement, le jeu de caractères ASCII ne comporte pas suffisamment de caractères de commande réservés aux spécifications (70 facteurs) de l'imprimante Epson FX-80. Pour y remédier, la plupart des effets sont alors produits en envoyant des « codes d'échappement » vers l'imprimante. Il s'agit de deux ou plusieurs codes de caractères commençant par le caractère ESC (ESCape / code ASCII 27). Ainsi, pour activer le mode spécial « espacement proportionnel » de l'Epson, vous enverrez le code ESC-p, c'est-à-dire le caractère d'échappement ESC suivi de la lettre p en minuscule.

Cela s'écrit en BASIC :

```
LPRINT CHR$(27);«p»
```

Le caractère ESC ne s'obtient pas de manière automatique en frappant la touche « ESC » de votre clavier; la fonction CHR\$ doit être utilisée. Sur le BBC vous ferez :

```
VDU2
VDU1, 27, 1, 112
VDU3
```

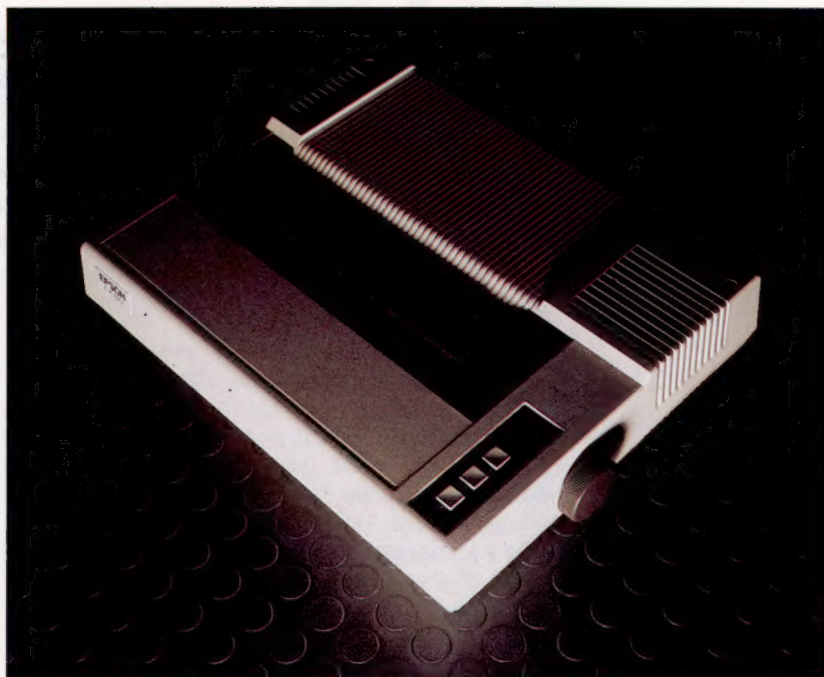
La commande VDU2 active l'imprimante; VDU1 signifie « envoyez le caractère suivant sur l'imprimante seulement » (PRINT dirigerait le caractère ESC également vers l'écran, ce qui aurait des conséquences fâcheuses). VDU3 inhibe l'imprimante.

Ces séquences de commande ne concernent que l'Epson FX-80. Si vous essayez d'envoyer la même séquence de code vers une autre imprimante, soit cela n'aura aucun effet, soit celle-là sera bloquée.

Création de facture

Notre deuxième listage de programme illustre l'utilisation de l'imprimante pour des en-têtes de factures. Les codes employés ici sont ceux de l'Epson FX-80. La gamme des imprimantes Epson est très répandue, au point que de nombreux fabricants de micro-ordinateurs s'assurent de la compatibilité Epson. Si votre imprimante n'est pas compatible, vous changerez les codes en conséquence.

```
999 REM EN-TETES DE FACTURES
1000 LPRINT CHR$(12)
1010 LPRINT CHR$(14);TAB(12);"GARAGE
DE LA BUTTE SARL"
1020 LPRINT CHR$(13);CHR$(13)
1030 LPRINT CHR$(27);"E";
1040 LPRINT TAB(36);
1050 LPRINT CHR$(27);"-" ;CHR$(1);
1060 LPRINT "FACTURE";
1070 LPRINT CHR$(27);"-" ;CHR$(0);
1080 LPRINT CHR$(27);"F";
1090 LPRINT CHR$(13);CHR$(13);CHR$(13)
1100 REM IMPRESSION DES DETAILS DE LA FACTURE
```



IAN MCKINNEL

Pour commencer, la ligne 1000 envoie le caractère de code 12 à l'imprimante. Il s'agit du caractère « saut de page » qui, comme son nom l'indique, place la tête de l'imprimante au début d'une nouvelle page. Vient ensuite le caractère ASCII 14, « caractère hors-code » qui suscite sur l'Epson l'impression agrandie des caractères qui le suivent. Il est utilisé dans notre programme pour l'en-tête, imprimant le nom du garage en gros. La fonction TAB sert à centrer l'en-tête.

CHR\$(13) est le caractère « retour-chariot »; il produit, lorsqu'il est imprimé seul, une ligne blanche. Plusieurs sont utilisés entre les lignes 1020 et 1090 pour faire ressortir le titre sur le haut de la facture. ESC-E à la ligne 1030 active le mode « gras » pour le texte qui vient après. La ligne 1050 active le mode « souligné » et la ligne 1070 l'inhibe après avoir imprimé et souligné le mot « Facture ». ESC-F inhibe le mode « gras ». La présentation de l'en-tête sera :

```
GARAGE DE LA BUTTE SARL

facture
```

Nous n'avons indiqué ici que le début du programme; un programme complet de présentation d'une facture supposerait des lignes de programme pour l'impression des détails : nom du client, marque de la voiture, somme due, etc. Ces détails proviendraient d'un ensemble de questions dont les réponses auraient été stockées comme variables.

Les deux programmes donnés ici ne sont que de simples exemples des utilisations possibles d'une imprimante à matrice. De nombreux usages nouveaux se répandent. En fait, la programmation de votre imprimante peut se révéler aussi passionnante que celle de votre ordinateur.

L'Epson FX-80
Imprimante très répandue pour les applications professionnelles et domestiques malgré son prix élevé. Le modèle FX-80 comporte une tête à 9 broches et a une vitesse maximale d'impression de 160 caractères par seconde. La plupart des progiciels (dont les traitements de texte) peuvent l'utiliser. (Cl. Ian McKinnell.)



Publipostage

Pour créer un système de publipostage avec un micro domestique, vous n'avez besoin que d'une imprimante, d'un programme de traitement de texte et d'un autre de base de données.

Le publipostage (mailing) informatisé peut aller beaucoup plus loin que la simple impression d'étiquettes d'adresses. En fait, si nous n'avions besoin que d'un outil pour écrire des adresses sur des enveloppes, un programme de base de données suffirait puisqu'il est très facile de constituer un fichier de noms et d'adresses. Chaque enregistrement de nom et d'adresse serait d'abord entré zone par zone puis, ce travail de saisie terminé, il serait possible d'utiliser le programme de base de données pour générer toutes les étiquettes désirées.

Puisque les programmes de base de données (de bonne qualité) disposent de certaines fonctions évoluées de sortie, il devrait être possible de sélectionner les étiquettes à imprimer selon certains critères : par exemple, toutes les adresses dont la zone « ville » est Paris.

Ce programme de publipostage se chargerait de la sélection et de l'écriture des étiquettes mais l'utilisateur aurait toujours à effectuer des tâches

ennuyeuses comme de taper les adresses sur chaque lettre, éventuellement. De plus, chaque lettre comporterait le nom et l'adresse d'un client, mais le texte de la lettre demeurerait anonyme et impersonnel. Il ne serait pas très pratique, également, de laisser dans le texte des espaces permettant d'insérer ultérieurement le nom de chaque client, puisque la longueur de chaque nom est variable. Vous devriez prévoir assez d'espace pour le nom le plus long à insérer, ce qui donnerait un aspect plutôt étrange à votre lettre « personnalisée ». Il serait donc nécessaire de la retaper autant de fois qu'il y a de noms dans la liste.

La solution idéale à ce problème serait de mettre sur pied un système où l'utilisateur pourrait composer une lettre type à l'aide de toutes les fonctions d'un programme de traitement de texte, et de demander à l'ordinateur d'extraire automatiquement dans une base de données les détails concernant chaque client afin d'ajouter des détails personnalisés dans chaque lettre.

Plusieurs possibilités sont offertes. L'une des plus simples est de se procurer un logiciel sur deux disquettes produit par Acorn — Memoplan et Fileplan, un programme de traitement de texte et un programme de base de données. Ces deux programmes font partie de la sélection de logiciels proposée par Acorn aux acheteurs d'un processeur Z80 installé sur le micro BBC modèle B.

Les programmes ne peuvent produire individuellement des lettres personnalisées et doivent être utilisés conjointement. Le fichier de noms et d'adresses est constitué par Fileplan et la lettre type est créée avec Memoplan. Chaque zone du fichier noms et adresses est numéroté, et l'utilisateur n'a qu'à taper aux endroits appropriés dans la lettre type le numéro de zone dont le contenu doit être inséré. L'ordinateur parcourt alors séquentiellement la liste de publipostage et fait apparaître dans la lettre type le contenu des zones concernées.

Le programme de traitement de texte ajuste automatiquement le texte adjacent afin que les destinataires reçoivent une lettre qui semble leur avoir été écrite personnellement. Memoplan permet à la personne qui rédige la lettre type d'inclure un pense-bête afin d'identifier le contenu d'une zone particulière. Par exemple, 2(NOM) indique que la deuxième zone renferme les noms des destinataires.

Les programmes spécialisés de publipostage destinés aux micro-ordinateurs de gestion, tels

Date
Date actuelle communiquée par le programme

Adresse du correspondant
Provenant de la base de données

Référence
A partir de la base de données

Nom
A partir de la base de données

Sujet
A partir de la base de données

Date
Date de correspondance à partir de la base de données

Lieu
A partir de la base de données

AIDA CONSTRUCTION
54, avenue Victor-Hugo
75016 Paris

Adresse du correspondant
Société MAZETTI
144, rue Emile Zola
75015 PARIS

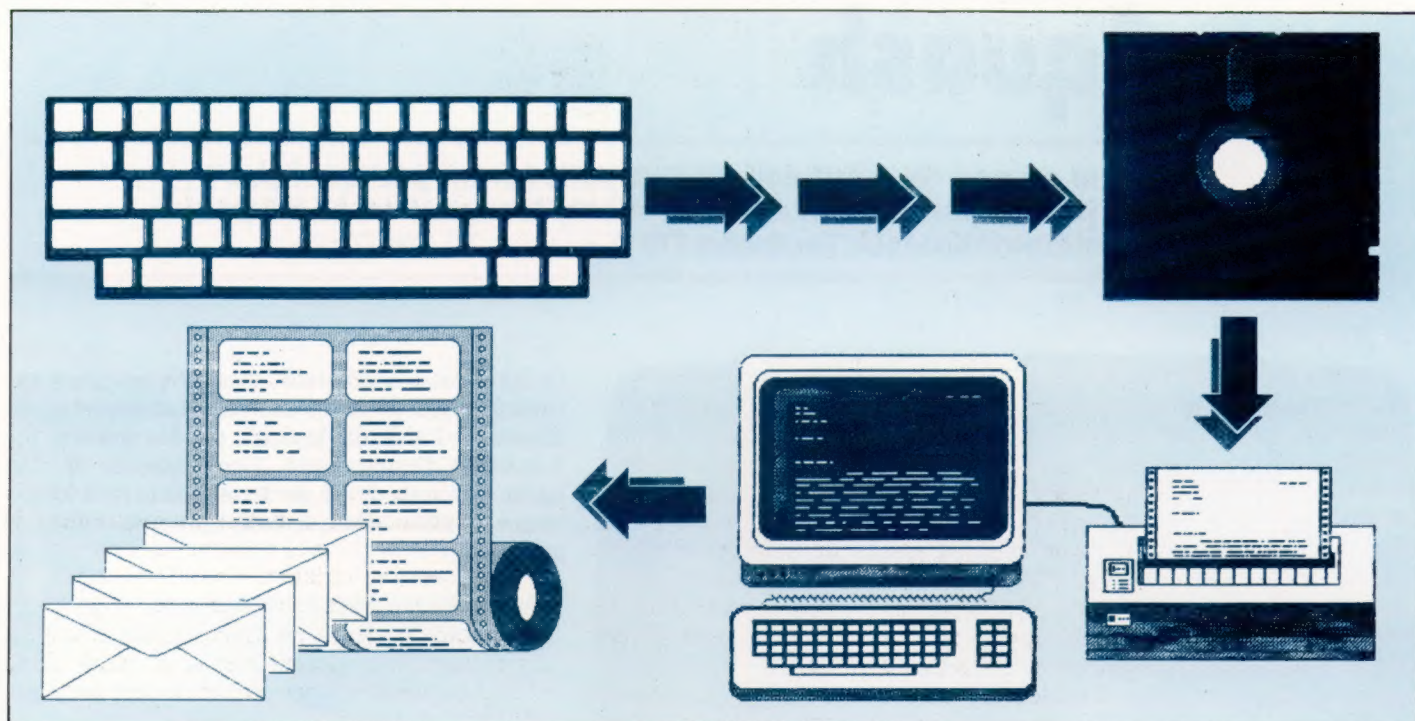
Référence
Vos références : WM/2546

Nom
Cher Monsieur Mazetti,

Sujet
Nous accusons réception de votre lettre du 4 juin 1984 nous invitant à soumettre un devis pour le contrat de la Place des Vosges. Nos service technique prépare actuellement un devis et il sera envoyé à vos bureaux de Paris d'ici un mois.

En vous priant d'agrèer, Cher Monsieur, nos meilleures salutations.

Jéan Cépède
Président



que Mailmerge de Micropro (le programme de publipostage le plus connu), comprennent de nombreuses fonctions pratiques et très évoluées. Il est non seulement possible de créer un grand nombre de fichiers d'adresses différents, mais ce type de programme peut également inclure des fonctions de recherche et de sélection très performantes qui permettent de choisir une portion précise d'une liste de publipostage.

La recherche et la sélection sont possibles grâce à des techniques standard d'indexation de base de données, et grâce à l'exécution de tests logiques sur les zones sélectionnées dans l'enregistrement.

Ces programmes évolués ont également des fonctions de mise en page très efficaces. Ces fonctions sont particulièrement pratiques, puisque les noms et adresses produits par le système peuvent être placés de manière à correspondre à la taille et à la forme des étiquettes. Avec le système Peachtree, par exemple, sélectionner l'option LABEL FORMAT du menu principal produit à l'écran l'image d'une boîte où apparaît une liste de toutes les zones des enregistrements de la liste. Cela donne à l'utilisateur une représentation visuelle de l'étiquette qui se crée. En plus, des fonctions servent à informer l'imprimante que la feuille d'étiquettes a une largeur équivalente à trois étiquettes afin d'en imprimer trois lors de chaque passage de la tête d'impression, et ainsi de suite.

Généralement, les programmes de publipostage sont surtout pratiques avec des listes de dimensions variables. Si vous ne devez taper que quelques étiquettes, il est probablement plus simple de les taper individuellement — sauf évidemment si l'information est déjà sur l'ordinateur! C'est pourquoi les programmes de publipostage pour ordinateurs domestiques

(même s'ils sont stockés sur ROM) vous obligent à stocker vos données sur disquette plutôt que sur cassette. La société britannique GCC commercialise depuis quelque temps son propre programme de base de données sur ROM nommé Starbase. Ce programme offre uniquement les fonctions de base de données. Mais cette société vient de lancer une version de Starbase livrée dans un module ROM de 16 K qui permet d'associer toutes les fonctions de publipostage avec le programme de traitement de texte Wordwise stocké en ROM lui aussi.

Starbase est livré avec un manuel et une disquette d'utilitaires. Utilisé avec Wordwise, ses fonctions de publipostage permettent de personnaliser des lettres types et d'effectuer la mise en page d'étiquettes. La disquette d'utilitaires permet d'envoyer des commandes à l'imprimante pendant l'exécution de l'option de mise en page d'étiquettes. Il est ainsi possible de sélectionner différentes polices de caractères avec un système équipé d'une imprimante adéquate.

Une des fonctions de publipostage de Starbase particulièrement utile est de pouvoir effectuer des opérations arithmétiques sur des zones dans des fichiers d'adresses. Il est ainsi possible de créer des factures personnalisées, laissant à l'ordinateur le soin d'effectuer tous les calculs individuels.

Les utilisateurs du Commodore 64 ont un excellent choix de programmes de publipostage. Parmi ceux-ci, mentionnons Visawrite, produit par Visa Software. Contrairement à Starbase, Visawrite peut fonctionner avec une disquette ou avec une cassette et peut être utilisé avec tout programme de base de données pouvant créer un fichier séquentiel. D'autre part, utilisé comme programme de traitement, il peut gérer jusqu'à 500 noms et adresses.

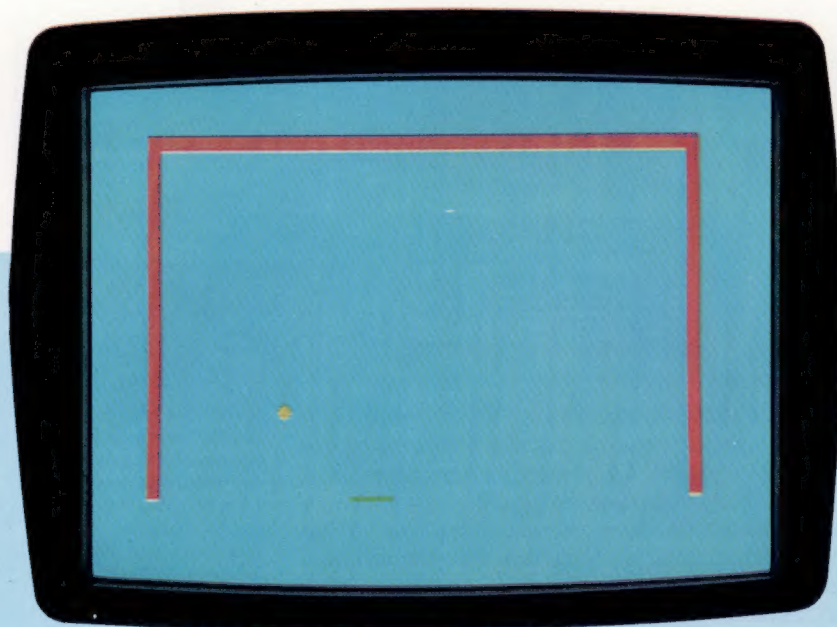
Travail à la chaîne

Le programme de traitement de texte produit une « lettre squelette » où il est possible d'insérer une information spécifique, comme des données, des noms et des adresses, tandis que le programme de base de données fournit les données associées à partir de ses enregistrements. Puis les adresses des lettres sont imprimées sur des étiquettes auto-collantes.

(Cl. Ian McKinnell.)

Squash

Faire un peu de sport devant son petit écran, pourquoi pas ?
Ce jeu qui paraît à été écrit par Pierre Monsaut en basic pour
le micro-ordinateur Thomson TO 7.



Grâce à votre ordinateur, vous pouvez jouer au squash, assis dans un fauteuil. La raquette se déplace à l'aide du joystick ou des touches Q, S et barre d'espace. Vous disposez de dix balles que vous devez garder en jeu le plus longtemps possible. Chaque balle renvoyée rapporte un point.

```

10 REM *****
20 REM * SQUASH *
30 REM *****
40 CLEAR ,,2
50 GOSUB 660
60 GOTO 180
70 D=2*((STICK(0)=7)-(STICK(0)=3))
80 IF D<>0 THEN DO=D
90 IF STICK(0)=0 THEN DO=0
100 RX=RX+DO
110 RETURN
120 D$=INKEY$
130 D=2*((D$="Q")-(D$="S"))
140 IF D<>0 THEN DO=D
150 IF D$=N$ THEN DO=0
160 RX=RX+DO
170 RETURN
180 LOCATE BX,BY
190 COLOR 3,6
200 PRINT N$;
210 BX=BX+DX
220 BY=BY+DY
230 LOCATE BX,BY
240 PRINT B$;
250 IF BY=22 AND ABS(BX-RX-3)>1 THEN 360
260 IF BY=22 THEN S=S+1:BEEP:DY=-DY
270 IF BY=1 THEN BEEP:DY=-DY
280 IF BX=2 OR BX=37 THEN BEEP:DX=-DX
290 ON JS GOSUB 70,120
300 IF RX<0 THEN RX=0
310 IF RX>33 THEN RX=33
320 LOCATE RX,RY
330 COLOR 2
340 PRINT R$;
350 GOTO 180
360 NB=NB+1
370 IF NB=11 THEN 480
380 LOCATE BX,BY
390 PRINT N$;
400 FOR I=1 TO 3
410 BEEP
420 FOR J=1 TO 100

```

```

430 NEXT J
440 NEXT I
450 DO=0
460 GOSUB 940
470 GOTO 180
480 LOCATE 13,5
490 COLOR 0
500 PRINT "SCORE :";S;
510 IF S>R1 THEN R1=S
520 LOCATE 13,10
530 PRINT "RECORD :";R1;
540 LOCATE 13,15
550 PRINT "UNE AUTRE ?";
560 NB=0
570 S=0
580 D$=INKEY$
590 IF D$<>" " THEN 580
600 D$=INKEY$
610 IF D$="" THEN 600
620 IF D$<>"N" THEN 50
630 SCREEN 4,6,6
640 CLS
650 END
660 CLS
670 SCREEN 1,6,6
680 ATTRB 1,1
690 DEFINT A-Z
700 DEFGR$(0)=255,255,255,0,0,0,0,0
710 DEFGR$(1)=24,126,126,255,255,126,126,24
720 N$=CHR$(32)
730 LOCATE 1,10,0
740 PRINT "JOYSTICK (0 ou N) ?";
750 D$=INKEY$
760 IF D$="" THEN 750
770 IF D$="0" THEN JS=1 ELSE JS=2
780 CLS
790 ATTRB 0,0
800 COLOR 6,1
810 FOR BX=1 TO 38
820 LOCATE BX,0
830 PRINT N$;
840 NEXT BX
850 FOR BY=1 TO 22
860 LOCATE 1,BY
870 PRINT N$;
880 LOCATE 38,BY
890 PRINT N$;
900 NEXT BY
910 R$=N$+N$+GR$(0)+GR$(0)+GR$(0)+N$+N$
920 B$=GR$(1)
930 RX=16
940 RY=23
950 BY=22
960 BX=INT(RND*34)+3
970 DY=-1
980 DX=(INT(RND*2)-0.5)*2
990 RETURN

```




D'une pierre deux coups

On peut bien sûr raccorder un micro-ordinateur à un poste de télévision; mais la qualité d'image est bien meilleure si l'on fait usage d'un moniteur.

La plupart des passionnés de micro-informatique se résignent assez vite aux couleurs indistinctes et aux images tremblotantes qu'ils obtiennent en branchant leur appareil sur un simple poste de télévision. Ceux qui ont assez de chance pour pouvoir se servir d'un moniteur n'en croient pas leurs yeux : les couleurs sont bien plus franches, l'image est stable et dépourvue de « scintillement » (ce tremblement si net aux marges du texte affiché sur l'écran). Il y a hélas un prix à payer, au propre comme au figuré : les moniteurs sont onéreux; ils ne peuvent capter les programmes de la télévision.

Mais il est désormais possible de combiner les avantages des deux systèmes. Le dispositif est assez simple : un téléviseur ordinaire pourvu d'une prise supplémentaire, sur laquelle on raccorde un micro-ordinateur pour avoir une image aussi bonne que sur un moniteur. Certains usagers en ont peut-être un, sans s'en douter, beaucoup d'appareils étant aujourd'hui équipés de connexions destinées au branche-

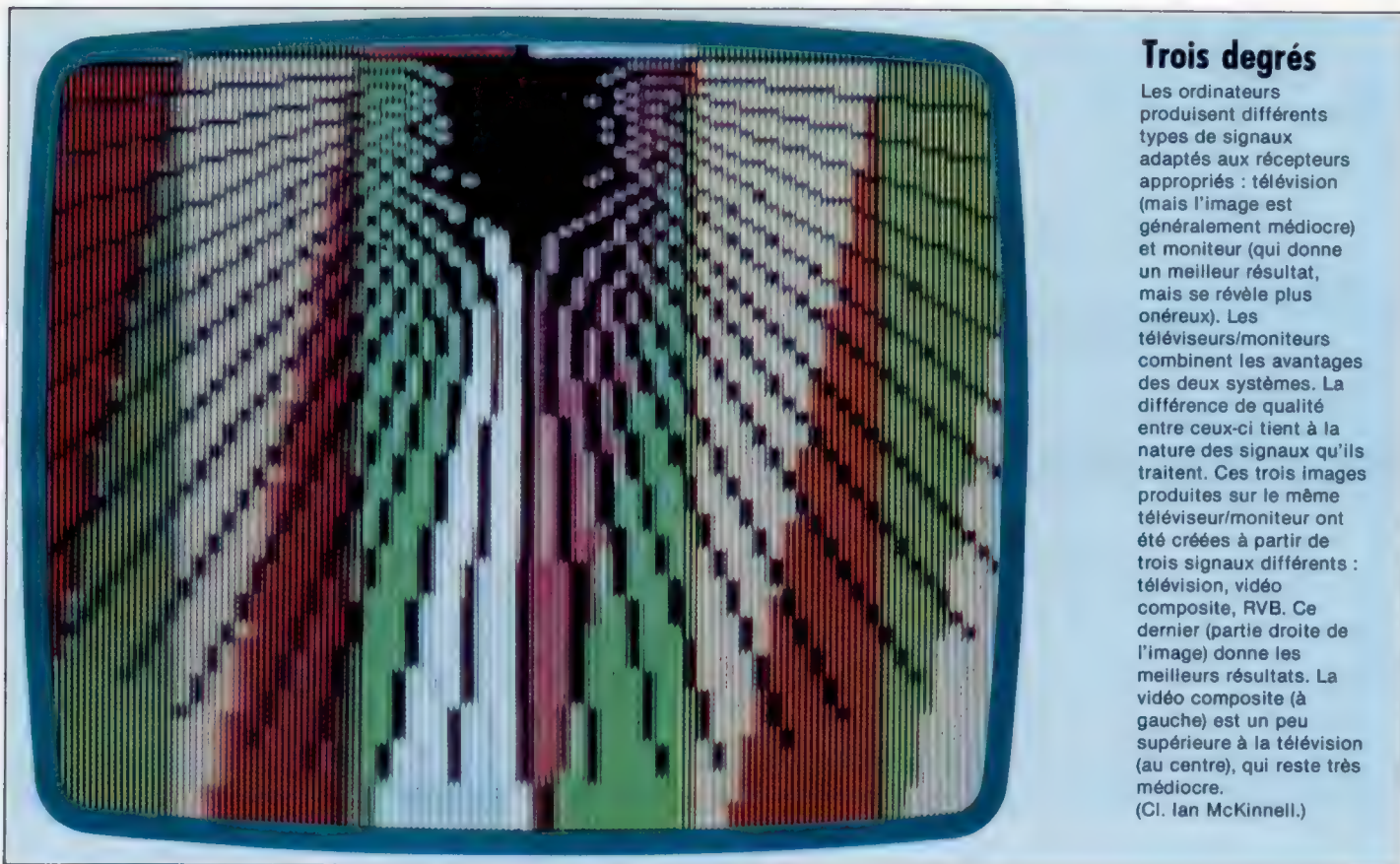
ment d'un magnétoscope, et dont on peut se servir pour y brancher un microprocesseur.

Le problème posé par les téléviseurs vient de la façon dont ils reçoivent leurs programmes. Ceux-ci sont transmis sous forme d'ondes radio captées par l'antenne et transformées en images. Un ordinateur domestique se borne à imiter ce processus en faisant passer ses messages à travers un modulateur (la petite boîte contenue dans ses flancs et qui se raccorde à l'antenne). Ce dernier fait du signal une onde radio que le récepteur de télévision peut accepter, mais qu'il modifie de nouveau au moment de l'affichage. Le signal en question peut donc être dégradé à deux reprises. Un moniteur, par contre, n'a pas besoin de modulateur et l'image, non transformée, sera évidemment bien meilleure.

Toutefois, si vous décidez d'en acheter un, n'oubliez pas de tenir compte des particularités de votre ordinateur. Les moniteurs font usage de deux grands types de signaux : RVB (rouge,

Trois degrés

Les ordinateurs produisent différents types de signaux adaptés aux récepteurs appropriés : télévision (mais l'image est généralement médiocre) et moniteur (qui donne un meilleur résultat, mais se révèle plus onéreux). Les téléviseurs/moniteurs combinent les avantages des deux systèmes. La différence de qualité entre ceux-ci tient à la nature des signaux qu'ils traitent. Ces trois images produites sur le même téléviseur/moniteur ont été créées à partir de trois signaux différents : télévision, vidéo composite, RVB. Ce dernier (partie droite de l'image) donne les meilleurs résultats. La vidéo composite (à gauche) est un peu supérieure à la télévision (au centre), qui reste très médiocre. (Cl. Ian McKinnell.)





vert, bleu) et de ce qu'on appelle vidéo composite. Tous deux permettent d'obtenir des résultats bien supérieurs à ceux d'un téléviseur; le signal RVB donne sans doute une meilleure image.

Par ailleurs, il existe aussi bien des postes de télévision modifiés pour accepter un signal moniteur, que des appareils spécifiquement construits en ce sens. Ces derniers sont généralement préférables; les autres subissent des transformations sans que le constructeur en soit informé, ce qui signifie qu'ils ne sont plus sous garantie. Les téléviseurs/moniteurs sont essentiellement prévus pour être employés avec des magnétoscopes. Ils fonctionnent avec un signal vidéo composite — cherchez une prise marquée « vidéo » ou « audiovisuel ». Les diagrammes qui accompagnent cet article vous montreront comment y connecter votre ordinateur (à condition que ce soit un modèle adapté). Une fois cela fait, réglez l'affichage de la même façon que vous choisiriez une chaîne.

Un des gros avantages de ce système est la présence du son (hauts-parleurs et amplificateurs), dont un moniteur standard est fréquemment dépourvu. C'est même pourquoi des micro-ordinateurs comme les Thomson, ou la gamme Commodore et Atari, se servent à cette fin de la télévision.

Si jamais votre appareil produit un signal RVB, votre choix sera plus limité. On peut citer trois modèles principaux de téléviseur/moniteur, le Sony Profeel, le ITT, ainsi que les postes pourvus de prises Péritel. Le premier accepte les deux types de signaux, mais fait usage d'un connecteur non standard. Celui d'ITT, de type RVB, est compatible broche à broche avec l'Oric et l'Atmos, tout comme avec certains autres ordinateurs. Depuis le 1^{er} janvier 1981, les téléviseurs vendus en France doivent être munis d'une prise Péritel. Le dispositif offre de nombreux avantages dans la mesure où derrière la prise, le câblage est effectivement réalisé. Il transmet aussi bien la vidéo composite que le RVB.

Même si les constructeurs de postes de télévision sont donc dans l'obligation d'équiper leurs appareils de prises de ce type, vérifiez si le vôtre en comporte une. Il est à noter que dans certains cas, on ne peut passer d'un mode à l'autre que par insertion et enlèvement de la prise, ce qui est plus gênant que de passer simplement d'un canal à l'autre. Toutefois, comme elle accepte des signaux d'entrée différents, il est possible de conserver le même téléviseur/moniteur si l'on vient à changer d'ordinateur.

Les résultats obtenus, en RVB comme en vidéo composite, sont d'ailleurs si remarquables que l'amateur d'informatique ne devrait pas hésiter — si bien sûr il en a les moyens — à faire l'acquisition d'un système de ce genre.

Le manque d'argent pour l'achat d'un moniteur ne doit pas constituer un frein pour celui qui désire faire ses premiers pas en informatique : la télévision familiale peut dans les débuts suffire.

Ferguson TX (RVB)

Ce téléviseur/moniteur accepte à la fois les signaux RVB et vidéo composite grâce à deux prises DIN. Le passage entre les trois modes d'affichage se fait à l'aide de boutons installés à l'avant de l'appareil, ce qui est sans doute la meilleure disposition possible : l'utilisateur peut ainsi suivre les programmes télévisés, ou se servir de son ordinateur, avec la plus grande facilité. Taille d'écran : 33 cm.

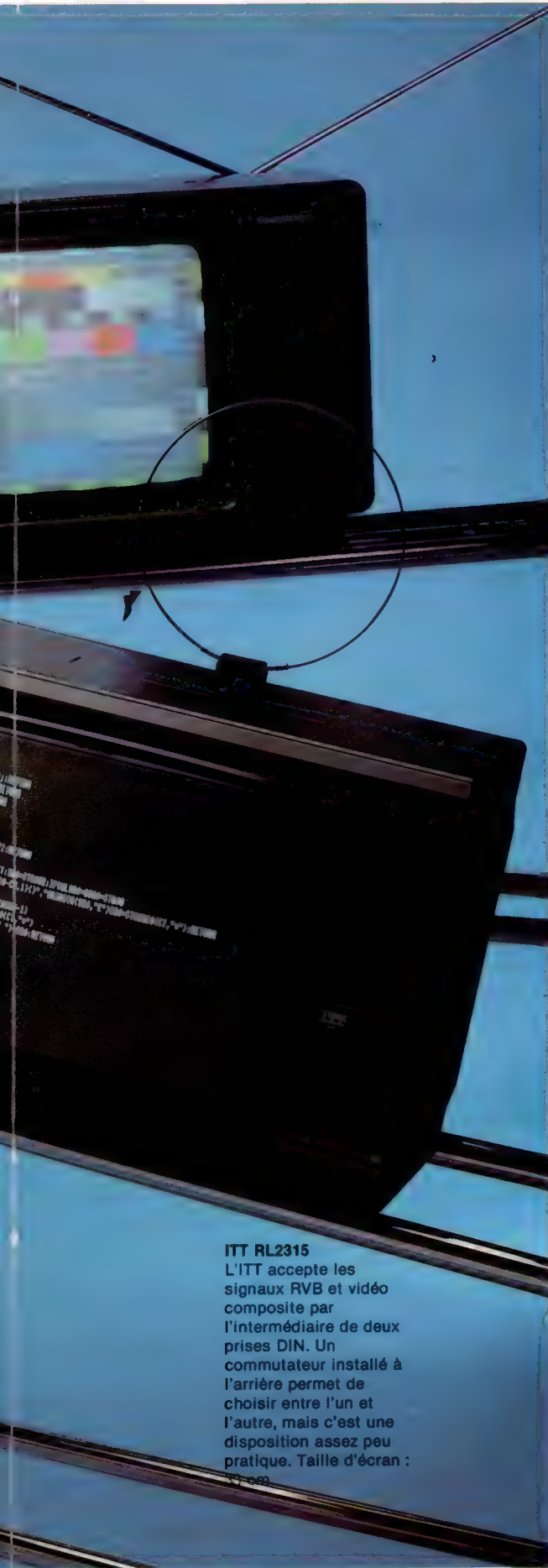
Normande 1534

Un des nombreux modèles de la gamme Normande, tous équipés de prises Péritel, et donc capables de traiter les signaux RVB et vidéo composite. Le modèle présenté ici a une taille d'écran de 33 cm, mais il existe en tout sept formats différents, qui peuvent être commandés manuellement ou à distance.



Fidelity CM14

Il existe en deux versions : l'une fait fonction de moniteur, grâce à une prise Péritel qui transmet les signaux RVB et vidéo composite. Taille d'écran, 33 cm. La seconde est un téléviseur/moniteur, et coûte donc un peu plus cher.



ITT RL2315
L'ITT accepte les signaux RVB et vidéo composite par l'intermédiaire de deux prises DIN. Un commutateur installé à l'arrière permet de choisir entre l'un et l'autre, mais c'est une disposition assez peu pratique. Taille d'écran :

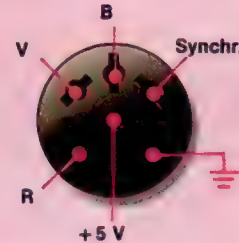
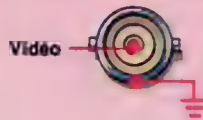
Chris Stevens

Avoir une bonne prise

Dragon (vidéo)



BBC/Electron (RVB et vidéo)



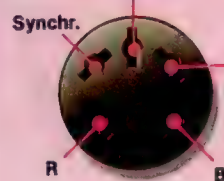
Lynx (RVB)



Commodore 64 Vic 20 (vidéo)



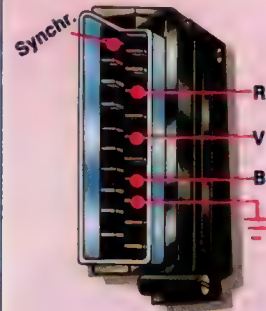
Oric (RVB)



Memotech (vidéo)



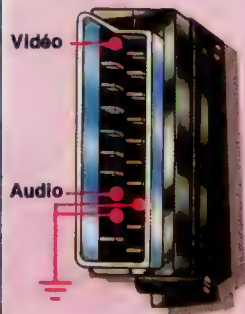
PÉRITEL pour RVB : FICHE



PÉRITEL pour RVB : PRISE



PRISE PÉRITEL POUR SIGNAL VIDÉO



Un moniteur peut fonctionner avec deux signaux différents : vidéo composite et RVB. Ce dernier se compose de signaux séparés (rouge, vert, bleu), plus un signal de synchronisation. Entrées et sorties passent généralement par des prises multibroches de type DIN. La vidéo composite mêle en un signal unique les couleurs et la synchronisation. Des prises phono ou BNC (à baïonnette) commandent le passage des entrées et des sorties, mais sur certains ordinateurs le son partage la même prise avec l'image, et il faut donc recourir à des multibroches.

Les diagrammes ci-contre vous montrent quelles sont les connexions propres à plusieurs micro-ordinateurs. Si votre téléviseur/moniteur est pourvu de branchements analogues, il vous suffira de joindre les deux prises à l'aide d'un câble, en respectant bien entendu les « mariages » indispensables (rouge et rouge, etc.).

Lorsque les téléviseurs/moniteurs sont équipés de prises PÉritel, le passage d'un mode à l'autre se fait généralement par l'intermédiaire de la prise elle-même, ce qui nécessite une alimentation en courant de 5 V (comme c'est le cas du BBC Micro) ou l'existence d'un commutateur.

Si certains ordinateurs sont absents de la liste, il existe néanmoins souvent une entreprise qui fabrique l'adaptateur nécessaire pour donner un signal RVB. Le dispositif se branche sur l'ordinateur.

Atari, Spectravidéo, le Vic 20 et la première version du Commodore 64 ont recours à la même méthode de branchement pour collecter le signal vidéo composite. Mais les modèles récents du 64 font usage d'une prise DIN à huit broches. La broche 2 est reliée à la terre, la broche 3 transmet le son et la broche 4 la vidéo. (Cl. Kevin Jones.)



Faire des vagues

Pour la plupart des utilisateurs de micro-ordinateurs, les mathématiques sont ennuyeuses. Pourtant, nous allons voir comment construire des courbes en trois dimensions en saisissant diverses fonctions mathématiques dans un programme.

La seule mention des fonctions sinus ou tangente suffit à donner des angoisses à ceux qui sont trop contents d'être débarrassés des mathématiques après leurs études. Pourtant, les mathématiques peuvent être amusantes!

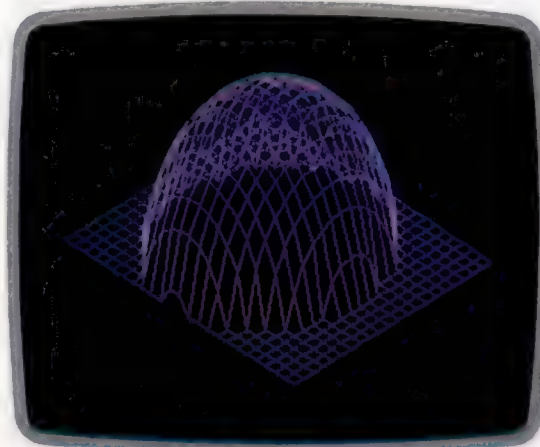
Les capacités graphiques de la plupart des ordinateurs personnels les prédisposent à l'affichage de courbes d'expressions mathématiques. Ces expressions peuvent produire d'étonnants motifs graphiques. Même ceux qui haïssent les maths peuvent trouver plaisir à générer leurs propres équations après avoir vu ces dessins.

Les motifs montrés ici ont été produits par un micro-ordinateur utilisant le programme listé. Ils ont tous été calculés en tant que courbes à trois dimensions. Nous savons ce qu'est une courbe à deux dimensions. Une courbe à trois dimensions se compose de plusieurs courbes à deux dimensions affichées simultanément, avec de légères différences de l'une à l'autre. Les ordinateurs ne pouvant afficher que des images en deux dimensions, le résultat n'est pas à proprement parler à trois dimensions, mais la disposition des images donne bien un effet de profondeur de champ.

Les programmes listés ici calculent les valeurs d'une équation à deux variables, X et Z. Le résultat Y est donné pour de nombreuses valeurs de X et de Z. Chaque valeur de Y sert à disposer un point à l'écran. Les valeurs possibles pour Y correspondent aux points sur l'axe vertical. Plus la valeur de Y est grande, plus le point tracé sera situé dans le haut de l'écran. Les points proches entre eux sont reliés par des lignes droites, ce qui donne un effet de courbe. Dans une direction, les courbes représentent des

équations de X et de Y, avec Z constant; les courbes qui viennent en intersection avec elles sont des équations de Y et de Z, avec X constant (elles sont alors représentées sur un plan avec les axes Y et Z, qui sont à angle droit avec le plan X-Y des courbes normales à deux dimensions). Ce mode d'affichage vous permettra d'appréhender des fonctions compliquées.

Il peut être amusant (et parfois difficile) de transcrire une équation sous forme de courbes. Pour modifier l'affichage, il faut changer la fonction de la ligne 170 du programme BASIC. Certaines fonctions peuvent être complexes et



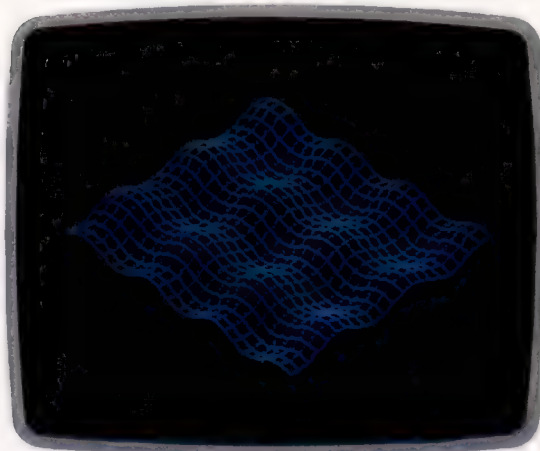
```
165 C=60-X*X-Z*Z
170 Y=SQR(C*(SGN(C)+1))/45
```

peuvent donc nécessiter plus d'une ligne de programme, auquel cas vous devrez utiliser toutes les lignes comprises entre 151 et 179.

Si vous devez choisir une fonction pour la belle forme graphique qui en résultera, vous devrez aussi veiller à ce que les valeurs produites ne sortent pas des limites de l'écran. Pour maintenir l'affichage dans ces limites, vous pourrez avoir recours à la division de la fonction par un nombre important.

Ce programme pourra fonctionner sur divers micros en en faisant des versions appropriées. Afin de permettre ces transcriptions, nous avons conçu le programme de sorte que la première partie initialise l'affichage de manière standard. Cela signifie qu'une équation convenant à une machine sera également appropriée pour une autre. La deuxième partie du programme sert à stocker les valeurs produites, en vue de tracer les points de l'image. Les résultats figurent dans un tableau. Ils sont assez longs à obtenir. Les calculs dépendent, en effet, de la

Création de formes graphiques
Modifiez le programme selon la formule mathématique indiquée en légende de chaque photographie, pour générer ces images en trois dimensions.
(Cl. Liz Heaney.)

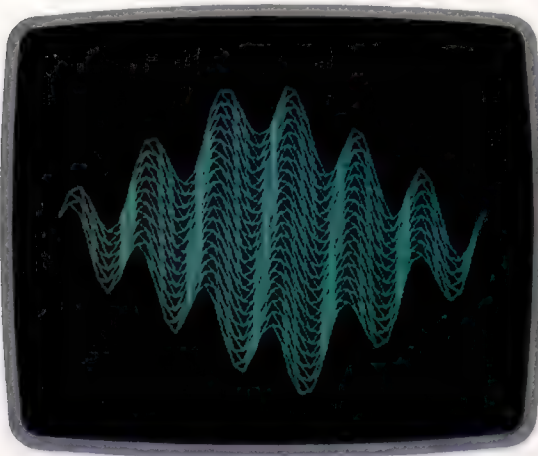


```
170 Y=(SIN(X)+COS(Z))/60
```


fonction choisie et peuvent prendre plusieurs minutes. Pendant ce temps, l'ordinateur semble ne rien faire. Le calcul préalable des fonctions

telles que SIN, COS, LOG, etc. D'autres graphiques seront dus à des fonctions propres aux ordinateurs; essayez à cet égard INT, SGN et ABS.

Vous pouvez améliorer le programme de plusieurs manières : faites par exemple que l'image produite corresponde automatiquement aux limites de l'écran ou changez de variables. Cela dit, même si vous utilisez le programme tel que, vous prendrez certainement plaisir à expérimenter les équations les plus farfelues qui vous passent par la tête. Les résultats vous surprendront sûrement.



170 Y=SIN(X+Z)/12

économise du temps sur l'ensemble des opérations. Si les calculs avaient lieu au fur et à mesure du tracé, le programme prendrait deux fois plus de temps pour créer l'image.

Nous avons listé plusieurs fonctions afin que vous les expérimentiez. Les figures montrent les résultats que vous pouvez obtenir. Essayez aussi de créer vos propres courbes en entrant dans le programme d'autres fonctions. Mais n'oubliez pas de rester dans les limites de l'écran. Vous devrez également veiller à ce que les opérations demandées soient possibles. Ainsi les erreurs les plus courantes sont, par exemple, d'essayer de diviser par zéro (ce qui donne un résultat infini), ou de demander la racine carrée d'un nombre négatif (ce qui n'existe pas).

Afin d'éviter la division par zéro, ajoutez systématiquement à une variable susceptible de prendre la valeur zéro, une constante très petite (par exemple, 0,00001). La seule façon de vous prémunir contre les racines carrées de nombres négatifs est d'utiliser la fonction ABS pour rendre tous les nombres positifs avant d'en extraire la racine.

Vous pourrez obtenir des images intéressantes avec de simples fonctions mathématiques



165 C=X*X+Z*Z+0.00001

170 Y=SGN(INT(23/C))/3+SGN(INT(55/C))/15

Programme en BCC basic

```

10 REM * TRACE DE GRAPHIQUE
20:
30 REM * INITIALISER ECRAN
40 TRAVERSEE=1280:HAUT=1024:HAUTEUR=-1
50 XINTERVAL=25:ZINTERVAL=15
60 LARGEUR=INT((TRAVERSEE/XINTERVAL)/2)
70 PROFONDEUR=INT(HAUT/ZINTERVAL/3)
80 MODE4:CLS:PRINTTAB(12)"CALCUL"
90:
100 REM * CALCUL DE L'IMAGE
110 START=20
120 DIM G(LARGEUR,PROFONDEUR)
130 FOR A=-PROFONDEUR/2 TO PROFONDEUR/2
140 FOR B=-LARGEUR/2 TO LARGEUR/2
150 X=A*20/LARGEUR:Z=B*20/PROFONDEUR
160 REM * INSERER LA FONCTION A LA LIGNE
    SUIVANTE
170 Y=(SIN(X)+COS(Z))/60
180 G(B+LARGEUR/2,A+PROFONDEUR/2)=Y*HAUTEUR
    +HAUT
190 NEXT B:NEXT A:CLS
200:
210 REM * TRACE DE L'IMAGE 1 X-Y PLAN
220 FOR Z=1 TO PROFONDEUR
230 XBASE=XINTERVAL*Z
240 ZBASE=HAUT/2+Z*ZINTERVAL+START+HAUTEUR
250 XANCIEN=XBASE+XINTERVAL
260 ZANCIEN=ZBASE-ZINTERVAL-G(1,Z)
270 FOR X=1 TO LARGEUR
280 XNOUVEAU=XBASE+X*ZINTERVAL
290 ZNOUVEAU=ZBASE-X*ZINTERVAL-G(X,Z)
300 PLOT 4,XANCIEN,ZANCIEN:PLOT 5,XANCIEN,ZNOUVEAU
310 XANCIEN=XNOUVEAU:ZANCIEN=ZNOUVEAU
320 NEXT X:NEXT Z
330:
340 REM * TRACE DE L'IMAGE 1 Z-Y PLAN
350 FOR X=1 TO LARGEUR
360 XBASE=XINTERVAL*X+PROFONDEUR*XINTERVAL
370 ZBASE=HAUT/2-X*ZINTERVAL+PROFONDEUR*ZINTERVAL
    +START+HAUTEUR
380 ZANCIEN=ZBASE-ZINTERVAL-G(X,PROFONDEUR-1)
390 XANCIEN=XBASE-XINTERVAL
400 FOR Z=0 TO PROFONDEUR-1
410 XNOUVEAU=XBASE-Z*ZINTERVAL
420 ZNOUVEAU=ZBASE-Z*ZINTERVAL-G(X,PROFONDEUR-Z)
430 PLOT 4,XANCIEN,ZANCIEN:PLOT 5,XNOUVEAU,ZNOUVEAU
440 XANCIEN=XNOUVEAU:ZANCIEN=ZNOUVEAU
450 NEXT Z:NEXT X
460:
470 REM * MAINTENIR L'AFFICHAGE
480 GOTO 470
    
```

Variantes de basic

Spectrum

Insérez LET dans les instructions d'assignation. Ajouter les lignes suivantes :

```

40 LET TRAVERSEE=256:LET HAUT=176:LET HAUTEUR=1
50 LET XINTERVAL=31:LET ZINTERVAL=3
60 CLS
230 PLOT 4,ZANCIEN,ZANCIEN:DRAW XNOUVEAU-ZANCIEN,
    ZNOUVEAU-ZANCIEN
410 PLOT 5,ZANCIEN,ZANCIEN:DRAW XNOUVEAU-ZANCIEN,
    ZNOUVEAU-ZANCIEN
    
```

Oric 1/Atmos

Ajouter les lignes suivantes :

```

40 *TRAVERSEE=229:HAUT=199:HAUTEUR=1
50 XINTERVAL=32:ZINTERVAL=4
60 CLS
200 CURSE* ZANCIEN,ZANCIEN:1:DRAW XNOUVEAU-ZANCIEN,
    ZNOUVEAU-ZANCIEN:1
430 CURSE* ZANCIEN,ZANCIEN:1:DRAW XNOUVEAU-ZANCIEN,
    ZNOUVEAU-ZANCIEN:1
    
```


Les trucs du métier

La plupart des utilisateurs de micro-ordinateurs ont simplement appris la programmation avec l'aide du manuel d'accompagnement. Mais, ainsi, ils ont peu de chances d'avoir appris quelques trucs.

Savoir bien programmer suppose beaucoup de pratique et d'expérience. Le programmeur novice acquerra, à la longue, la vision claire et simple de l'expert sur un problème de programmation, et saura d'emblée trouver l'approche la plus directe pour le résoudre. Malgré l'importance de la pratique, il est tout à fait possible de faire bénéficier l'utilisateur novice de l'expérience de ses devanciers. Cette série de cours doit vous y aider. Nous commençons par des généralités qui constituent les connaissances de base. Les étapes de la programmation sont bien connues : l'énoncé clair et complet du problème en termes pratiques, puis son étude de plus en plus détaillée, et ce, jusqu'à sa formulation précise. La description elle-même du problème comporte presque toujours l'énoncé de sa solution principale. Celle-ci doit alors être explicitée et se transformer en une méthode effective de traitement. En programmation, seule la dernière étape devrait concerner l'écriture de code qui suit rigoureusement l'application des principes énoncés précédemment. Lorsque la phase d'écriture du programme empiète sur la recherche des solutions, le résultat est à coup sûr médiocre avec un code de peu d'efficacité.

On appelle généralement les solutions « algorithmes », ou « processus de traitement du problème analysé par étapes logiques ». Les performances du programme dépendent principalement de l'algorithme, qui doit être complet et rigoureux. Le programme devra sur cette base être à même de répondre à toutes les situations envisageables en entrée du traitement. Ses capacités théoriques et pratiques de résoudre par sa logique interne toutes les configurations possibles de cas ne doivent jamais être mises en défaut. Il est inutile de dire que cette perfection idéale est rarement atteinte et vérifiée. Mais c'est là l'objectif premier et ce critère doit être présent dès le départ du développement d'un programme.

Les solutions trouvées doivent être fiables autant qu'exhaustives et justes dans tous les cas de figures. Elles doivent non seulement traiter les problèmes pour lesquels elles ont été élaborées, mais encore pouvoir intégrer de manière prévisible et sûre des situations nouvelles. Cela signifie généralement reconnaître des conditions d'erreurs potentielles et être capable d'interrompre le traitement sans endommager les données. En outre le programme devra informer son utilisateur par des messages de statut. Il n'est pas facile de dire qu'un programme est

vraiment fiable, par contre l'expérience montre souvent qu'un programme a des failles.

La réalisation de programmes complets, cohérents et puissants est un noble idéal qui entre généralement en conflit avec un objectif non moins nécessaire, l'économie de moyens. Tout coûte, ne serait-ce que le temps que vous passez à écrire des programmes. Il vous faudra toujours à un moment ou à un autre vous décider à cesser de travailler sur un programme suffisamment abouti et passer à l'étude d'un nouveau projet. Et même si votre temps était illimité, la mémoire de votre ordinateur et sa vitesse de traitement ne le sont pas. Il se peut qu'en cherchant à entourer l'algorithme central de nombreuses précautions, tests et protections, vous passiez davantage de temps à rendre le programme fiable qu'à le développer.

Tests et dépistage d'erreurs

Lorsque les premières erreurs de syntaxe et de logique ont été corrigées, le temps est venu de tester le programme. C'est quelque chose de tellement familier qu'il n'y a pas lieu d'insister. Et pourtant c'est un processus assez mal compris de manière générale. En effet, pour un programme un tant soit peu complexe, le nombre de combinaisons possibles de conditions d'entrée au traitement est tel, qu'il est tout à fait impossible d'envisager des tests exhaustifs. Aussi la phase de tests devra-t-elle se consacrer aux aspects les plus sensibles, aux points forts du programme. Les professionnels considèrent qu'il n'existe pas de programmes parfaits, mais simplement de mauvais tests.

Lorsque les tests se révèlent efficaces, ils mettent au jour les inadéquations du programme. Ils doivent le faire en outre de manière logique, afin que le dépistage des erreurs prenne aussi peu de temps que possible. Comme les tests, ce dernier aspect n'est souvent pas satisfaisant dans la mesure où il est difficile de le conduire efficacement. Le dépistage d'erreurs doit donc être abordé comme un problème distinct, avec énoncé du projet, analyse, algorithme et tests. Dans la pratique, c'est davantage une plaie dont il faut se débarrasser au plus vite (et au hasard), avec ce que cela peut comporter de conséquences aléatoires.

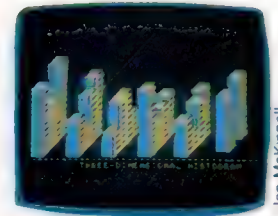
Avec le développement du programme et de ses diverses phases, la satisfaction éprouvée à le voir prendre forme génère un effet pervers dans l'esprit du programmeur. Il pense que son pro-



gramme marche, qu'il « tournera » toujours bien et qu'il n'y aura jamais lieu de lui apporter des modifications. Le code, bien sûr, est un modèle de clarté... Si les programmeurs ont besoin de documentation dans leur programme, le programme, lui, n'en a pas besoin. Aucun programme ne parle de lui-même, et il est absolument nécessaire de pouvoir « rentrer » dans un programme en cours d'exploitation. Il y aura toujours lieu en effet de le modifier de temps en temps. Comme tout autre mécanisme, il a besoin de maintenance, ce qui suppose des modes d'emploi. Aussi les programmes doivent-ils être documentés de manière interne (lignes de

commentaires REM à l'usage du programmeur) et de manière externe pour l'utilisateur.

Autant de leçons que les programmeurs sur gros ordinateurs ont dû apprendre un jour, et qui sont ignorées par ceux qui abordent la programmation sur micro-ordinateurs. Il s'agit globalement de la nécessité de structurer la programmation par une approche univoque et complète de l'analyse (qui ne se contente pas de règles de prudence comme d'éviter les « GOTO » pour des « WHILE...WEND »). Les programmes puissants sont écrits par des programmeurs efficaces sur la base d'une expérience structurée et d'une réflexion logique.



Ian McKinnell

Tableau à barres
Les couleurs et profondeurs (relief) des barres du tableau sont facilement paramétrables en modifiant les valeurs des variables appropriées.

```

239 REM*****
400 REM* TABLEAUX A BARRES A 3 DIM. *
401 REM*****
500 GOSUB 1500: REM INITIALISER
720 YY=22:XX=2: REM COORDONNEES D'ORIGINE
760 GOSUB 3200: REM TRACER LES AXES
800 FOR E=LT-1 TO 0 STEP -1
820 OC=XX+E*DB:OL=YY-E*DB
940 GOSUB 2200: REM ENTRER LES DONNEES
960 FOR D=1 TO NN
920 HT=DT(D)
940 X0=OC+(D-1)*(BB+LT*DB):Y0=OL-HT*DB
960 GOSUB 4000: REM AFFICHER LA BARRE
980 NEXT D
1000 NEXT E
1100 XP=10:YP=23:GOSUB 3500
1120 PRINT"MIET. A 3 DIMENSIONS
1200 A$=INKEY$:IF A$="" THEN GOTO 1200
1400 END
1499 REM*****
1500 REM* INIT. S/R *
1501 REM*****
1520 CL$=CHR$(167): REM EFFACER ECRAN
1540 PRINT CL$
1560 PO$=CHR$(19): REM<RETOUR>CURSEUR
1580 RT$=CHR$(13): REM RETOUR
1600 BB=2:DB=1: REM DIMENSIONS BARRE
1620 SW=40:SD=25: REM DIM. ECRAN
1640 HB=SD-DB: REM HT MAX. BARRE
1660 DIM B$(HB*DB)
1680 FOR K=1 TO SD:PO$=PO$+RT$:NEXT K
1800 DIM DT(SW)
1900 GOSUB 2400: REM TRACER BARRE
2100 LT=4: REM FACTEUR PROFONDEUR
2190 RETURN
2199 REM*****
2200 REM* ENTRER DONNEES DU TABLEAU S/R *
2201 REM*****
2220 READ NN
2240 FOR Z=1 TO NN:READ DT(Z):NEXT Z
2310 DATA 6,12,10,4,7,8,10
2320 DATA 5,7,8,9,6,7
2330 DATA 6,7,8,9,5,3,9
2340 DATA 5,11,6,4,11,6
2390 RETURN
2399 REM*****
2400 REM* TRACER TOUTE LA BARRE S/R *
2401 REM*****
2500 TC$=CHR$(158): REM COTES=JAUNE
2520 FC$=CHR$(31): REM DEVANT=BLEU
2540 RV$=CHR$(18): REM INVERSION ON
2560 NL$=CHR$(146): REM INVERSION OFF
2580 CH$=CHR$(29): REM CURSEUR A DROITE
2600 CS$=CHR$(32): REM CAR. ESPACE
2620 C1$=CHR$(169): REM CAR. " "
2640 C2$=RV$+C1$: REM CAR. " "
2680 FOR K=1 TO SW
2700 SP$=SP$+CH$
2720 RC$=RC$+CR$
2740 FF$=FF$+CH$
2760 NEXT K
2800 TL$=SP$+ "/"
    
```

Structure Modulaire et suffisamment auto-documentée.

Documentation
Cette routine est certainement importante, mais totalement inexplicquée.

Noms de variables
De nombreux commentaires mais pas assez explicites.

Exhaustivité/exactitude
Les résultats sont-ils valables pour toutes valeurs? Sont-ils justes dans tous les cas — pour une valeur négative par exemple?

Dépistage d'erreurs
Pas de bit de parité ni de gestion des erreurs.

Documentation
Bonne : pas de nombres magiques (non expliqués), ni de caractères de contrôle mystérieux; tout est intelligible.

Du bon et du mauvais
Ce programme qui affiche des tableaux à barres en trois dimensions mélange le meilleur et le pire. La documentation interne est bonne lorsqu'elle existe et la structure est modulaire. Mais le programme n'est pas auto-documenté; il n'y a pas de dépistage des erreurs.

```

2820 BL$=SP$+NL$+C1$
2840 FL$=GAUCHE$(FF$,BB)
2900 L$=TC$+C2$+RV$+DROITE$(TL$,BB)
2920 FOR K=1 TO DB
2940 B$(K)=GAUCHE$(RC$,DB-K)+L$+GAUCHE$(SP$,K-1)
2960 NEXT K
3000 L$=FC$+RV$+FL$+TC$+DROITE$(TL$,DB)
3020 FOR K=DB+1 TO HB
3040 B$(K)=L$
3060 NEXT K
3100 L$=FC$+RV$+FL$+TC$
3120 FOR K=1 TO DB
3140 B$(HB+K)=L$+DROITE$(BL$,DB+2-K)
3160 NEXT K
3190 RETURN
3199 REM*****
3200 REM* TRACER AXES S/R *
3201 REM*****
3300 PRINT TC$: REM ATTRIBUER COULEURS
3320 FOR Y=2 TO YY-1
3340 XP=XX-1:YP=Y:GOSUB 3500:PRINT"|"
3360 XP=XX+YY-Y:GOSUB 3500:PRINT"/"
3380 NEXT Y
3400 YP=YY
3420 FOR X=XX-1 TO SW-1
3440 XP=X:GOSUB 3500:PRINT"-"
3460 NEXT X
3490 RETURN
3499 REM*****
3500 REM* AFFECTER CR$R @ XP, YP S/R *
3501 REM*****
3600 PRINT LEFT$(PO$, YP)TAB(XP-1)
3620 RETURN
3999 REM*****
4000 REM* AFFICHER UNE PARTIE DE BARRE S/R *
4001 REM*****
4100 FOR V=1 TO HT
4120 XP=X0:YP=Y0+V-1
4140 REM***** REM PLACER (CR$R)
4160 PRINT B$(V)
4180 NEXT V
4200 FOR V=1 TO DB
4220 XP=X0:YP=Y0+HT+V-1:GOSUB 3500
4240 PRINT B$(HB+V)
4260 NEXT V
4490 RETURN
READY.
    
```

Variantes de basic

Ce programme a été écrit en basic Microsoft et devrait être applicable sans modifications aux micros d'affichage 25 x 40. Les dimensions de l'écran sont initialisées à la ligne 1620. Les valeurs pour le caractère de contrôle, initialisées aux sous-programmes figurant aux lignes 1500 et 2400, concernent le Commodore 64.

La vallée des Trolls

Les jeux d'aventures permettent à qui s'y adonne d'accomplir des exploits héroïques dans des univers fantastiques. Le joueur doit faire usage de tout son esprit. Examinons Twin Kingdom Valley, de la firme Bug-Byte.

Les jeux d'aventures, tels qu'ils existent sur ordinateur, viennent des jeux de rôles du type Donjon et Dragon. Au début des années soixante, certains programmeurs de gros systèmes mirent au point les premières versions, se servant de la très vaste mémoire de leurs appareils pour recréer des mondes imaginaires, mais très détaillés, et peuplés de sorciers, de monstres et de nains. Les jeux d'aujourd'hui en sont les descendants, bien qu'ils se soient beaucoup diversifiés.

Toutes les bonnes aventures, cependant, ont un point commun : le joueur doit avoir l'impression d'évoluer dans un univers réel, et de s'investir entièrement dans les situations qu'il affronte, comme s'il devenait un personnage de roman. Les premiers jeux ne comportaient que du texte, mais certains exemples récents intègrent des graphismes haute résolution, dans un souci de réalisme accru. Le Hobbit, d'après le célèbre roman de J.R.R. Tolkien, fut le premier grand succès en ce domaine. Nous examinerons ici Twin Kingdom Valley, un jeu d'aventures graphique au cadre classique : forêts, châteaux médiévaux.

Les puristes sont très réservés face à la présence de graphismes dans une aventure : c'est un attrait supplémentaire, mais qui, pour eux, ne sert souvent qu'à camoufler un certain manque d'imagination du programmeur. L'intrigue de Twin Kingdom Valley reste extrêmement simple. Le joueur devient une sorte d'errant qui s'aventure dans une vallée où règnent deux rois

en guerre : celui du Désert, celui des Bois. La contrée est parcourue de rivières, qui se jettent dans le lac magique Watersmeet. Le héros doit accumuler autant de trésors que possible. Une fois que cela est fait et que son score atteint 1024 points, il se passe quelque chose de très inattendu — mais nous ne gâcherons pas votre plaisir en vous révélant quoi...

Taper des instructions au clavier permet de contrôler actions et déplacements. Le programme reconnaît vingt-trois verbes, qui se combinent à des noms (ceux des objets présents dans le jeu). On peut demander à l'ordinateur de tenir compte d'une phrase comme « Frapper garde avec marteau ». Il faudra naturellement que vous ayez bel et bien un marteau, et qu'un garde soit en vue ! Les huit directions dans lesquelles vous pouvez vous déplacer correspondent à celles qu'indiquerait une rose des vents ; vous pouvez également aller vers le haut ou vers le bas. Bien entendu, vous n'êtes pas seul, et d'autres personnages parcourent le royaume. Demandez-leur, au besoin, de vous donner ce qu'ils possèdent ; mais prenez garde, presque tous se jetteront sur vous dès que vous chercherez à leur parler !

Les illustrations évoquent cent soixante-quinze endroits différents. Elles varient avec les versions. Le BBC Micro, par exemple, consomme beaucoup d'espace mémoire en graphisme haute résolution ; la plupart de ses images sont donc composées d'éléments de base diversement combinés ; le même motif d'arbre, répété et modifié à chaque fois, forme une forêt. Le Commodore 64 a une mémoire plus importante, et une certaine capacité d'animation, grâce à ses objets graphiques qui permettent de faire courir des écureuils, ou de voir passer des nuages dans le ciel... Il est d'ailleurs possible de faire disparaître les dessins, ce qui accroît le rythme du jeu.

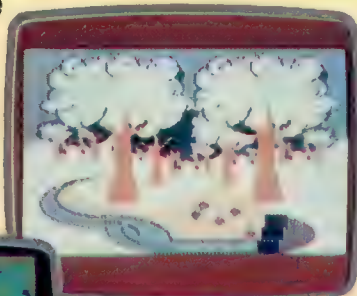
On peut se demander, toutefois, s'il n'aurait pas mieux valu accroître la complexité du jeu, et sacrifier un peu au graphisme.

Les royaumes de l'expérience

Le BBC Micro fait essentiellement usage de combinaisons d'éléments graphiques de base, tandis que le Commodore 64, grâce à sa mémoire plus vaste et son emploi d'objets graphiques, peut se permettre plus de variété et même quelques moments d'animation.



BBC Micro



Commodore 64

Twin Kingdom Valley : BBC Micro, Commodore 64.

Éditeurs : Bug-Byte Software.

Auteur : Trevor Hall.

Manche à balai : inutile.

Format : cassette.



Tracé de pixels

Dans ce premier article d'une série explorant les applications graphiques, regardons l'usage du langage d'assemblage 6502 pour accéder à l'écran haute résolution du Commodore 64.

Les diverses étapes et procédures impliquées dans les graphiques haute résolution du Commodore ont déjà été examinées; il faut d'abord commuter la puce d'interface vidéo (VIC) sur le mode haute résolution et modifier le pointeur d'adresse de base du jeu de caractères; le bloc de 8 K commençant à l'adresse 8192, qui comportera la table d'implantation mémoire de l'écran, doit être remis à zéro; et la table d'implantation mémoire écran normale (adresses 1024 à 2033 — \$0400 à \$07E7), qui doit maintenant servir à l'information couleur de l'écran, doit être initialisée.

La première partie de notre programme s'occupera donc des deux tâches importantes qui doivent être menées à bien avant de passer en haute résolution: l'information couleur doit être mise à chaque emplacement normal d'écran, et la table d'implantation de 8 K doit être remise à zéro.

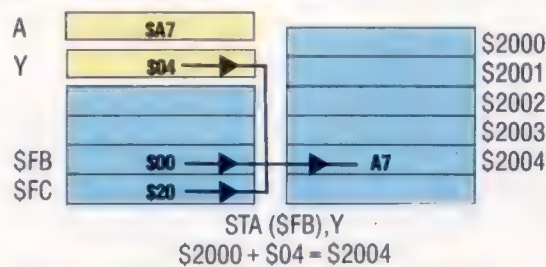
Pour permettre l'appel au même sous-programme en entrant ou en sortant du mode haute résolution, un drapeau appelé HRSFLG sera utilisé. De plus, il se peut que nous ne voulions pas toujours remettre à zéro la table d'implantation en entrant dans le mode « haute résolution », surtout si nous désirons laisser une forme précédemment tracée à l'écran. Si nous voulons effacer l'écran, nous utiliserons un second drapeau appelé CLRFLG. L'organigramme montre comment seront utilisés ces deux drapeaux dans le sous-programme en langage machine.

Considérons maintenant la tâche assez simple consistant à accéder à des blocs de mémoire de 256 octets ou moins à l'aide de boucles en langage machine. Le fragment de programme suivant place le nombre \$03 dans chaque emplacement à partir de l'adresse BASE jusqu'à BASE+255 (soit au total 256 emplacements) en utilisant l'adressage indexé absolu.

```
LDY $00
LDA $03
NEXT STA BASE, Y
DEY
BNE NEXT
```

Remarquez qu'en utilisant cette technique, on accède d'abord à BASE, mais pour le reste du bloc on commence par BASE+255 pour descendre jusqu'à BASE+1. Notre programme exige d'accéder à plus de 256 octets de mémoire, et pour cela nous devons utiliser une autre forme d'adressage connue sous le nom d'indirect post-indexé. Cette méthode utilise la page zéro pour calculer

des adresses en n'importe quel endroit de la mémoire. La plus grande partie de la page zéro est utilisée par le système d'exploitation du Commodore 64, mais quelques octets libres sont réservés pour l'usage du programmeur en langage machine. Deux de ces octets sont 251 et 252 (\$FB et \$FC). Dans cette méthode d'adressage, l'ordinateur suppose que l'octet lo de l'adresse est spécifié dans l'emplacement de la page zéro et l'octet hi dans l'emplacement suivant. Ainsi, des instructions telles que STA (\$FB),Y où \$FB et \$FC contiennent \$00 et \$20, et Y contient \$04, calcule l'adresse voulue comme suit :

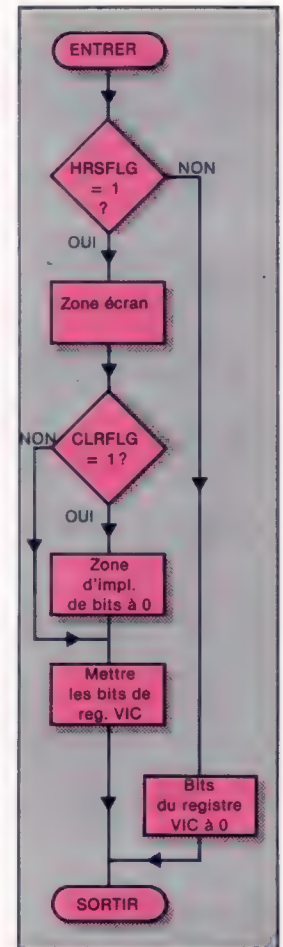


Afin d'accéder à tout un bloc mémoire de 256 octets, on peut utiliser une méthode analogue à celle qui vient d'être décrite. La puissance de cette méthode d'adressage réside dans notre capacité à accéder à l'octet hi de l'adresse BASE. En incrémentant cet octet hi de un, on augmente en fait l'adresse BASE de 256 (c'est-à-dire le bloc mémoire suivant). Nous pouvons appliquer cette technique aux tâches consistant à placer des informations couleurs dans les zones écran normales et effacer la zone de table d'implantation de bit.

La zone écran va de \$0400 à \$07E7. Cela signifie qu'elle consiste en trois blocs de 256 octets et un reste de \$E7 octets. Le bloc de langage d'assemblage étiqueté « zone écran couleur » (voir pages suivantes) fait usage de l'adressage indirect post-indexé pour placer l'information couleur dans chaque octet. On utilise les variables SCBLO et SCBHI — les octets lo et hi de l'adresse de début d'écran normale — et SCBLK et SCREM — le nombre de blocs de 256 octets et le reste, respectivement. PTR est l'emplacement page zéro utilisé pour stocker l'octet lo de l'adresse de base.

Calcul des positions de pixels

La seconde partie de notre sous-programme en langage machine concerne le calcul du bit dans la zone d'implantation du bit correspondant à des coordonnées (x, y) données. Pour l'écran



Routine HIRSES

Cette routine haute résolution accomplit trois tâches différentes. Elle prépare l'information de l'écran couleur, met à zéro la zone d'implantation de bit, prépare et remet à zéro les bits de registre HIRSES (haute résolution) — selon l'état des drapeaux HRSFLG et CLRFLG.

haute résolution, les valeurs de x sont comprises entre 0 et 319, et celles de y entre 0 et 199. Un octet suffit pour toutes les valeurs possibles de y , mais il faudra deux octets pour stocker les valeurs de x supérieures à 255. En BASIC, étant donné les coordonnées x et y , le bit correspondant est calculé par le fragment de programme suivant :

```
1000 HB=XAND248:VBYTE=INT(Y/8)
1010 RMY=YAND7:RMX=XAND7
1020 ROW=VBYTE+320+HB
1030 BYTE=BASE+ROW+RMY
1040 POKEBYTE, PEEK(BYTEOR(2+7-RMX))
```

Le sous-programme en langage machine correspondant doit effectuer les mêmes calculs. HB, ROW, BASE et BYTE requièrent chacun deux octets, ce qui rend plus complexes les processus arithmétiques. La plupart du codage est évident, mais il y a deux sections intéressantes : celle où y est divisé par 8 et celle où VBYTE est multiplié par 320. La division par des puissances de deux s'effectue facilement :

```
45 = 00101101
45/2 = 22 = 00010110
22/2 = 11 = 00001011
11/2 = 5 = 00000101
5/2 = 2 = 00000010
2/2 = 1 = 00000001
1/2 = 0 = 00000000
```

Chaque fois qu'on effectue une division par deux, les bits sont décalés d'une position vers la droite. La division par huit équivaut donc à trois décalages logiques vers la droite ou LSR (Logical Shift Right). Comme nous n'avons besoin que de la partie entière de la solution, nous pouvons ignorer tous les bits qui « tombent » à droite du nombre. L'opération LSR place le bit perdu dans la retenue, de sorte que nous pourrions l'utiliser si nous voulions. La multiplication par deux correspond, bien évidemment, à un décalage d'une position vers la gauche ou ASL. Nous pouvons utiliser ce fait pour créer une routine qui multiplie VBYTE par 320. 320 peut être écrit comme 5×64 , et 64 lui-même est $2 \times 2 \times 2 \times 2 \times 2 \times 2$. Ainsi, si nous prenons VBYTE, lui additionnons cinq fois sa propre valeur, puis effectuons six ASL, nous l'avons effectivement multiplié par 320. Le seul inconvénient est que la solution peut fort bien être supérieure à 255 et qu'il nous faudra donc deux octets.

Enfin, les deux routines que nous avons examinées sont toutes deux appelées à l'intérieur d'un programme BASIC utilisant SYS, et il est important que, lorsque les sous-programmes en langage machine (S-P M/CODE) sont terminés, le programme BASIC puisse continuer. Pendant l'exécution d'un programme BASIC, l'interpréteur utilise les registres X, Y et A. Comme ces registres sont également employés par les sous-programmes en langage machine, il est important de stocker leur contenu lors du passage en langage machine, puis de les restituer en quittant le sous-programme.

```
1 GOTO 200
2 POKE 53265, PEEK(53265)AND223
3 POKE 53272, PEEK(53272)AND2400RA
4 STOP
200 REM **** C64 HI-RES DEMO ****
210 :
220 POKE 56,32:CLR: REM LWR MEMTOP
250 GOSUB 3000: REM S-P INITIALISE
350 :
360 REM **** MODES HIRES ****
370 :
380 PRINT CC$:PRINT :PRINT
390 INPUT"COULEUR AVANT-PLAN":FG
400 INPUT"COULEUR ARRIERE-PLAN":BG
410 TT=FG+16+BG: REM CALC. COLOUR
420 POKE COLOUR, TT: REM COL TO M/C S-P
430 POKE HRSFLG, 1: REM SET HIRES ON
440 POKE CLMFLG, 1: REM CLRHIRES SCRAN ON
450 SYS BEGIN: REM ENTER M/C S-P
460 :
500 REM **** DESSIN ****
510 :
515 Z=0:Y1=50:Y2=150:X1=150:SP=6
520 FOR Y=Y1 TO Y2 STEP SP
530 FOR X=X1 TO X2 STEP SP
540 GOSUB 1000: REM PLOT POINT
550 NEXT X
555 Z=Z+SP
557 NEXT Y
560 :
565 GETJ$:IF J$<>" " THEN 565
570 GETA$:IF A$="" THEN 570:REM ATTEND TOUCHE PRESSEE
580 :
600 REM **** EFFACER ECRAN HIRES ****
605 POKE HRSFLG, 1:POKE CLMFLG, 1
610 SYS BEGIN
620 :
630 REM **** LINE DEMO ****
640 X1=0:X2=300:Y1=0:Y2=190:SP=1
670 GOSUB 1500: REM LINE PLOT
680 :
685 GETJ$:IF J$<>" " THEN 685
700 GETA$:IF A$="" THEN 700:REM ATTEND TOUCHE PRESSEE
710 :
720 REM **** RESTAURE ECRAN ****
730 :
740 POKE HRSFLG, 0: REM HRES OFF
750 SYS BEGIN
760 PRINT CC$:PRINT :PRINT
770 PRINTTAB(9)"****FIN DE PROGRAMME****"
780 END
999 :
1000 REM **** HIRES PLOT S-P ***
1010 :
1020 XHI=INT(X/HX):XLO=X-XHI*HX
1030 POKE XBYTE,XLO:POKE XPAGE,XHI:POKE YBYTE, Y
1035 SYS PLOT: REM ENTRE PLOT S-P
1040 RETURN
1500 REM *** LINE PLOT S-P ***
1550 CS=(Y2-Y1)/(X2-X1):CB=CS*X1-Y1
1600 FOR X=X1 TO X2 STEP SP
1650 Y=X+CS-CB
1700 GOSUB 1000: REM PLOT POINT
1750 NEXT X
1800 RETURN
3000 REM *** INITIALISATION ***
3020 CC$=CHR$(147): REM CLEAR SCRAN
3025 HX=256
3030 HRSFLG=49408: REM %C100
3040 CLMFLG=49409: REM %C101
3050 COLOUR=49410: REM %C102
3060 XBYTE=49411: REM %C103
3070 XPAGE=49412: REM %C104
3080 YBYTE=49413: REM %C105
3085 BEGIN=49422: REM %C10E
3090 PLOT=49539: REM %C103
3095 PRINT CC$:PRINT:PRINT
3100 PRINTTAB(9)"****CHARGEMENT M/CODE****"
3150 PRINTTAB(9)"1) M/CODE SUR BANDE"
3200 PRINTTAB(9)"2) M/CODE EN DATA"
3250 PRINTTAB(9)"3) M/CODE EN MEN"
3300 PRINT"TAPE NUMERO D'OPTION"
3350 FOR LP=0 TO 1 STEP 0
3400 GET OP$
3450 IF OP$="0" AND OP$>"4" THEN LP=1
3500 NEXT LP
3600 ON VAL(OP$) GOSUB 4000,5000,6000
3900 RETURN
4000 REM** CHARGER M/CODE DE LA BANDE"
4100 PRINT "INSERER BANDE CONTENANT S-P LANGAGE MACHINE"
4200 IF A$="" THEN A=1:LOAD "PLOTSUB.HEX", 1, 1
```




```

4900 RETURN
5000 REM** CHARGER M/CODE DU S-P DATA"
5005 PRINTTAB(11)"**CHARGEMENT**"
5010 FOR I=HRSFLG TO HRSFLG+312:READ A
5020 POKE I,A:S=S+A:NEXT I
5030 READ CC:IF CC<>S THEN PRINT"ERREUR DE
TOTAL":END
5040 DATA 2,0,255,255,2,2,255,255,2,18
5050 DATA 255,255,2,2,72,138,72,152,72
5060 DATA 173,0,193,240,83,169,0,133,251
5070 DATA 169,4,133,252,162,3,160,0,173
5080 DATA 2,193,145,251,136,208,251,238
5090 DATA 252,202,48,8,208,244,145,251
5100 DATA 160,231,208,238,173,1,193,240
5110 DATA 24,169,0,133,251,169,32,133
5120 DATA 252,162,32,160,0,169,0,145,251
5130 DATA 136,208,251,238,252,202,208
5140 DATA 246,173,24,208,41,240,9,8,141
5150 DATA 24,208,173,17,208,9,32,141,17
5160 DATA 208,76,125,193,173,24,208,41
5170 DATA 240,9,4,141,24,208,173,17,208
5180 DATA 41,223,141,17,208,104,169,104
5190 DATA 170,104,96,72,138,72,152,72
5200 DATA 173,4,193,141,7,193,173,3,193
5210 DATA 41,248,141,6,193,173,3,193,41
5220 DATA 7,141,8,193,173,5,193,41,7,141
5230 DATA 18,193,162,3,78,5,193,202,208
5240 DATA 258,173,5,193,141,9,193,169,0
5250 DATA 141,11,193,141,12,193,162,5
5260 DATA 173,11,193,24,109,9,193,141,11
5270 DATA 193,202,208,243,162,6,14,12
5280 DATA 193,14,11,193,144,3,238,12,193
5290 DATA 202,208,242,173,11,193,24,109
5300 DATA 6,193,141,11,193,173,12,193
5310 DATA 109,7,183,141,12,193,173,11
5320 DATA 193,24,105,0,141,11,193,173,12
5330 DATA 193,105,32,141,12,193,173,11
5340 DATA 193,24,109,10,193,141,11,193
5350 DATA 173,12,193,105,0,141,12,193
5360 DATA 173,11,193,133,251,173,12,193
5370 DATA 133,252,169,1,141,13,193,56
5380 DATA 169,7,237,8,193,170,14,13,193
5390 DATA 202,208,250,160,0,177,251,13
5400 DATA 13,193,145,251,76,125,193
5410 DATA 38698:REM*TOTAL*
5500 RETURN
6000 REM** MCODE DEJA DANS S-P:MEM **
6100 RETURN

```

L'utilisation de PLOTSUB

Le programme basic de démonstration montre les diverses étapes impliquées dans l'utilisation des routines haute résolution (HIRES) en langage machine.

1. Si vous avez un assembleur, vous pouvez taper le programme en langage d'assemblage, l'assembler, le sauvegarder comme fichier source, puis sauvegarder le code objet entre \$C100 et \$C238 sous le nom PLOTSUB.HEX. N'essayez pas d'exécuter le sous-programme à ce stade, car l'écran haute résolution écraserait probablement l'assembleur lui-même en écrivant par-dessus.

2. Pour utiliser les routines haute résolution dans un programme, vous devez abaisser MEMTOP (voir ligne 220 du programme) et charger le code à partir de la bande (voir sous-programme 4000).

3. Il est aussi possible de sauvegarder le sous-programme 5000 comme un programme basic (appelé MCODELOAD, par exemple). Lorsque vous voudrez l'utiliser, abaissez MEMTOP, puis chargez et faites tourner MCODELOAD (chargeant ainsi le langage machine en mémoire). Tapez NEW, puis chargez le programme que vous voulez faire tourner — les routines haute résolution sont maintenant en mémoire et on y accède par les instructions SYS adéquates.

4. La dernière donnée dans le sous-programme 5000 est un total de contrôle — somme de toutes les données précédentes. Si le programme s'arrête en affichant le message ERREUR DE TOTAL, c'est que vous avez mal entré les données, et il faut corriger la faute avant de continuer.

```

|*****|
|*****EXIT M/C*****|
EXIT PLA
TAX
PLA
TAX
PLA
RTS
|*****HIRES PLOT CALCUL *****|
PHA
TXA
PHA
TYA
PHA
|*****CALCUL HORIZ. BYTE*****|
LDA XHI
STA HBHI
LDA XLO
AND #0FB
STA HBLO
LDA XLO
AND #07
STA REMX
|*****CALCUL VERT. BYTE*****|
LDA YLO
AND #07
STA REMY
LOX #03
SHIFT LSR YLO
DEX
BNE SHIFT
LDA YLO
STA VBYTE
|*****CALCUL RANGE*****|
LDA #00
STA ROWLO
STA ROWHI
LDX #05
LDA ROWLO
CLC
ADC VBYTE
STA ROWLO
DEX
BNE FIVE
LDX #06
ASL ROWHI
ASL ROWLO
BCC NCARRY
INC ROWHI
DEX
BNE MULT
|*****ADD HORIZONTAL BYTE*****|
LDA ROWLO
CLC
ADC HBLO
STA ROWLO
LDA ROWHI
ADC HBHI
STA ROWHI
|*****ADD HIRES MAP BASE*****|
LDA ROWLO
CLC
ADC #MPBLO
STA ROWLO
LDA ROWHI
ADC #MPBHI
STA ROWHI
|*****ADD REMAINDER OF YLO*****|
LDA ROWLO
CLC
ADC REMY
STA ROWLO
LDA ROWHI
ADC #008
STA ROWHI
|*****COMBINE 2 BYTES D'*****|
|*****ADRESSE SUR PAGE ZERO*****|
LDA ROWLO
STA PTR
LDA ROWHI
STA PTR+1
|*****CALCUL PIXEL POSN.*****|
LDA #01
STA BPOS
SEC
LDA #07
SBC REMX
TAX
ASL BPOS
DEX
BNE POWER
|*****TURN ON PIXEL*****|
LDY #00
LDA (PTR),Y
ORA BPOS
STA (PTR),Y
JMP EXIT
|*****PASSER EN MODE HIRES*****|
|*****ET EFFACER ZONE BIT ***|
PHA
TXA
PHA
TYA
PHA
LDA HRSFLG
BEQ RESET
|*****ZONE ECRAN COULEUR*****|
LDA #SCBLO
STA PTR
LDA #SCBHI
STA PTR+1
LDX #SCBLK
LDY #00
LDA COLOUR
AGAIN STA (PTR),Y
DEY
BNE AGAIN
INC PTR+1
DEX
BHI CLTEST
BNE AGAIN
STA (PTR),Y
LDY #SCREM
BNE AGAIN
CLTEST LDA CLRFLO
BEQ HRESON
|*****EFFACER ZONE BIT*****|
LDA #MPBLO
STA PTR
LDA #MPBHI
STA PTR+1
LDX #MPBLK
LDY #00
LDA #00
NEXT STA (PTR),Y
DEY
BNE NEXT
INC PTR+1
DEX
BNE NEXT
|*****PASSER EN MODE BIT*****|
HRESON LDA #0010
AND #0F0
ORA #008
STA #0018
LDA #0011
ORA #020
STA #0011
JMP EXIT
|*****REPASSER A L'ECRAN NORMAL**|
RESET LDA #0018
AND #0F0
ORA #004
STA #0018
LDA #0011
AND #0DF
STA #0011

```




Une nouvelle ERE

A l'image de ce qui se passe aux États-Unis ou en Grande-Bretagne, les entreprises d'édition de logiciels se multiplient en France. Pour ERE Informatique, les débuts sont prometteurs.



Emmanuel Viau
Dynamique créateur d'entreprise, Emmanuel Viau est un passionné d'informatique, de science-fiction et de vidéo-clips. Son aventure commerciale est un modèle du genre. Souhaitons que les jeunes informaticiens soient nombreux à l'imiter.

De bons programmes
ERE informatique, c'est avant tout une gamme d'une vingtaine de programmes originaux, rédigés pour cinq micro-ordinateurs différents. Les premières étapes dans la constitution d'une vaste bibliothèque de logiciels français.

Parmi les éditeurs de logiciels destinés au grand public, les grands noms sont souvent associés à des grandes entreprises d'électronique ou de l'édition de livres. Hachette et Nathan, Thomson et Matra, tout le monde connaît. Pourtant, elles ne sont pas seules à vouloir occuper le terrain de plus en plus doré de ce marché. Dans des bureaux (de fortune parfois), on crée, on produit et on édite des logiciels destinés à l'éducation, aux jeux, à la gestion personnelle, et cela à Paris, à Toulouse, à Lyon ou à Nancy.

Pour la plupart de ces petites entreprises qui sont nées avec la première grande percée des « octets » dans les foyers, le téléphone arrive souvent en même temps que le visiteur! Le cliché n'est pas déplacé. Emmanuel Viau ne le cachait pas lorsqu'il ouvrait, il y a moins d'un an, un local dans le 8^e arrondissement de Paris, à l'enseigne de « ERE Informatique ».

Cet homme de 24 ans découvre l'informatique pendant son service militaire et, avec sa solde, il achète un ZX 81. Il est touché par la grâce de l'informatique. Retourné à la vie civile, il suit un stage accéléré chez Control Data et se met rapidement à écrire des programmes de jeu. Mais à l'époque, en France, le système éditorial n'avait pas encore saisi l'intérêt de la création informatique. Bref! Se sentant seul, Emmanuel Viau décide de créer sa propre société. L'argent vient des copains et le salon de l'appartement familial sert de premier local... Peu importe, l'idée est d'abreuver en français les ordinateurs domestiques qui commencent à envahir le marché, à l'instar de ce qui se passait déjà aux États-Unis ou en Grande-Bretagne. Bien sûr, il n'est pas question dans l'immédiat de faire des scores de vente aussi élevés que dans ces pays, où la barre des 100 000 cassettes pour un même logiciel est souvent dépassée.

Mais on sait chez ERE Informatique s'entourer aujourd'hui de bons auteurs pour connaître de bons succès de vente. Un véritable club (à but lucratif) a d'ailleurs été créé. Et plus de dix créateurs s'y retrouvent avec comme seules obligations d'être imaginatifs et d'avoir quelques connaissances techniques. Car pour créer un logiciel, il n'est pas forcément nécessaire d'être un informaticien professionnel.

Quoi qu'il en soit, les royalties d'un auteur de logiciel tournent autour de 15 % du chiffre d'affaires réalisé sur son produit. Il n'est donc pas rare de voir certains auteurs obtenir des gains supérieurs à 20 000 francs pendant plusieurs mois de suite. Un tel résultat suscitera peut-être des vocations nouvelles. Emmanuel Viau, comme d'autres éditeurs, l'espère. Pour qu'une entreprise comme ERE Informatique soit rentable, aucun des domaines de l'informatique familiale n'est privilégié ou rejeté.

Si les auteurs n'ont à se consacrer qu'à la création, l'entreprise doit de son côté assurer toute la logistique, c'est-à-dire tout le processus de fabrication depuis la maquette jusqu'à la duplication en passant par la conception de la jaquette et du mode d'emploi. Ce dernier point, essentiel pour le futur utilisateur, est trop souvent oublié par les auteurs.

Reste alors la partie commerciale, la vente aux grossistes et aux importateurs, avec l'espoir toujours présent — même non exprimé —, qu'un logiciel va devenir numéro 1 au hit-parade des ventes avec la fortune à la clé. Une nouvelle ère est vraiment arrivée.



PROGRAMME N° 25

PLAN DE FINANCEMENT

Du sérieux cette semaine avec un programme de financement de vos emprunts, moyennant la donnée « taux d'intérêt » facile à obtenir auprès de votre banque. Ce programme vous permettra de manière très aisée de calculer tous les paramètres nécessaires à l'évaluation correcte d'un emprunt.

L'écriture de ce programme est tout à fait standard et ne devrait poser aucun problème, quel que soit l'appareil utilisé.

Remarque. Le symbole « ^ », qui se lit « puissance », est celui de l'élevation à une puissance n .

Exemple : $2 \wedge 4 = 2^4 = 16$.

```
50  REM plan de financement
60  REM la banque prête un capital K
70  REM à un taux d'intérêt I % remboursable
80  REM en N années, avec des échéances
90  REM constantes.
100 ?
105 REM saisie du capital prêté en francs
110 INPUT "capital prêté : .....": K
115 REM saisie de la durée du prêt en années
120 INPUT "durée du prêt en années : .....": N
125 REM saisie du taux d'intérêt en %
130 INPUT "taux d'intérêt en % : .....": I
135 REM saisie de l'année de départ du prêt
140 INPUT "année de départ du prêt : .....": P
150 REM R est la valeur du capital à rembourser
160 R = K
165 REM mise en valeur décimale du taux
170 I = I / 100
175 REM calcul de l'annuité constante en francs
180 U = (K * I) / (1 - 1 / ((1 + I) ^ N))
185 REM écriture des résultats
190 ? "CAPITAL:";K;"ANNUITE:";U
200 REM
210 ? : ?
220 ? "ANNEE. AMORTISS. VALEUR RESIDUELLE"
230 ?
230 REM boucle sur le nombre d'année du prêt
240 FOR X = 1 TO N
250 REM calcul de la part de l'amortissement M.
260 M = U - (R * I)
265 REM calcul de l'année correspondant au remboursement
    en cours
270 A1 = P + X
275 REM calcul de la valeur résiduelle du capital K
    réactualisé après le remboursement.
```



```

280 R = R - M
285 REM écriture des résultats : année en cours A 1
      amortissement M et le capital résiduel à rembourser R
290 ? A1 ; HTAB 8 ; ? M ; HTAB 20 ; ? R
300 NEXT X
310 REM fin du programme
320 END

```

Une fois le programme écrit, taper RUN au clavier; le programme demande alors la valeur du capital devant être emprunté ainsi que les conditions du prêt : les lignes 110 à 140 s'affichent sur l'écran, l'une après l'autre, après chaque introduction des valeurs. Sur l'écran apparaît ensuite le contenu de la ligne 190 donnant le montant du capital K et celui de l'annuité U.

Ensuite, le contenu de la ligne 220 s'affiche donnant les titres des colonnes : années de remboursement, l'amortissement, le capital restant à rembourser (valeur résiduelle), ainsi qu'une succession de plusieurs lignes donnant ces valeurs (une ligne par année de prêt). Faire RUN éventuellement pour relancer le programme.

ORGANIGRAMME

